Volume 5, Issue 5 May, 1983
$1.50

TRS-80® Microcomputer News

- Model 4
- Home Issue

- PC-2 Assembly Language—Part 3





Fort Worth Scene

READER SURVEY

Some time back I mentioned that we took a reader survey. Well, the results are in and are neatly summarized in a very large book on my desk. We thought you might like to see some of the results.

Most of our readers have Model I or III computers. The Model I/III owners are followed by Color Computer owners, then Model II and Pocket Computer Owners. No real surprises in this ranking, especially if you consider how new Pocket Computers were when we did the survey.

Overall, 48% of the respondents rated *Microcomputer News* as being "excellent" or "very good". And 66% of the respondents considered the magazine to be just right in regard to technical content. There was a large indication that you want direct information about products, as well as more information on programming and some additional articles on advanced/technical material.

All of this information is being considered as we make plans for future issues. We are developing an overall content strategy, which we think will meet most of the needs and desires that have been identified. Our objective will be to balance material over a year's worth of issues since it is almost impossible to balance each issue.

NEW PRODUCTS

I am writing this in mid March, which means that most of you have just received the March issue. We began introducing you to six new TRS-80s in the February issue with the Model 12. March gave you the PC-4, April the Model 100, and in this issue we are introducing the Model 4. June and July will also introduce new machines to finish out the six introductions that were announced early in 1983.

I would appreciate hearing your reaction to the introduction articles that we ran and are running. We feel that it is important to get this information to you as soon after the product introductions as possible, and our goal has been to provide much more information in *Microcomputer News* than you will typically read in the other magazines. What I need to know is, how well are we filling your desire to know about these new products? Did we emphasize what you wanted to read about, or did we miss the mark.

We hope that you will find the May, June and July issues much more user oriented than some of our recent issues have been. There is so much information, and so few pages, that it is hard to decide what to print and what to leave out. We thought you might like a few issues filled with user items, and that is what we hope to give you in these three issues.

In closing, I would like to say that I am really excited about the Model 4 and especially excited about the Model III to Model 4 upgrade. As a consumer I appreciate the fact that my investment (time and money) in Model I hardware and software (yes, I am one of the many who are still operating Model Is) is reasonably transportable to both Models III and 4. I also appreciate that for the price of adding drive zero to a cassette based Model III I can move from a 32K Model III to a 64K Model 4 with Model III, TRSDOS, LDOS and CP/M compatibility.

MAGAZINES

Below are four magazines of special interest to TRS-80 owners that we believe have editorial content of high quality and will be of use to our customers.

80-US (Covers all TRS-80's)
3838 South Warner Street
Tacoma, WA 98409

Color Computer Magazine
Highland Hill
Camden, ME 04843

Rainbow (Covers the TRS-80 Color Computer.)
5803 Timber Ridge Dr.
Prospect, KY 40059

two/sixteen magazine
P.O. Box 1216
Lancaster, PA 17603



TRS-80 Microcomputer News
Volume 5 Issue 5
MAY 1983

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. Copyright 1983 by Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. All rights reserved.

Reproduction or use, without express written permission from Tandy Corporation, of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than Ivan Sygoda's Profile article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new full size TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Certain smaller TRS-80 Microcomputers will not include this free subscription. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars ($12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1.50 suggested retail each.

The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price $9.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007,535.

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

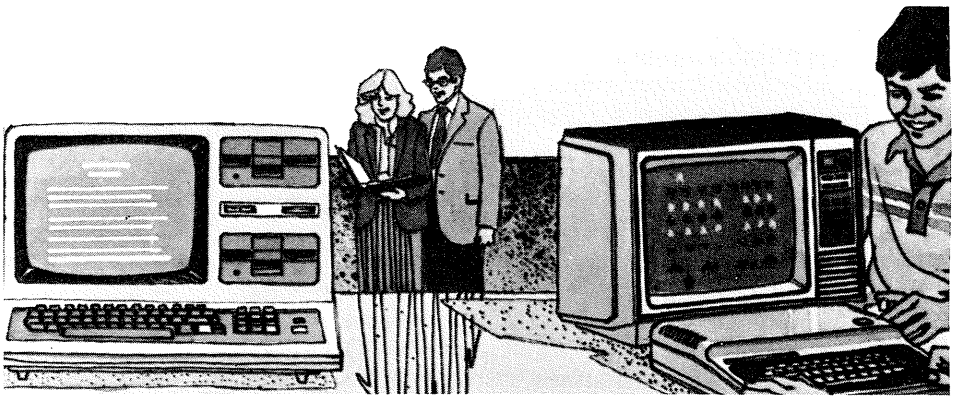
CompuServe™	CompuServe, Inc.
CP/M®	Digital Research
Dow Jones™ NEWS/RETRIEVAL Service®	Dow Jones & Co., Inc.
VisiCalc®	VisiCorp, Inc.
XENIX™	Microsoft
Program Pak™	Tandy Corporation
SCRIPSIT™	Tandy Corporation
TRSDOS™	Tandy Corporation
TRS-80®	Tandy Corporation

TRS-80® Microcomputer News

Contents:

Color Computer
Programs
Character Recognition by Bill Robinson . . . 43
Disk Directory by Bery Rinaldo . . . 42
Disk Editor and Disassembler by Ashok Basargekar . . . 40
Homebudget by Richard Neill . . . 44
Notepad Calculator by James McCracken . . . 44
OKZAP by Dennis Tanner . . . 39
Communications Corner by Al and Dru Simon . . . 10
Computer Clubs . . . 6
Computer Customer Service . . . 5
For Dot Addressable Printers
Databases
CompuServe . . . 7
A Variety of Games and Entertainment Features
Education
Creating a Network 2 Color Logo Lab by Dennis Tanner . . . 15
Educational Color Computer Users Network . . . 9
A Grading Technique using Scripsit by Timothy Little . . . 13
Fort Worth Scene by Bruce Elliott . . . 2
General Interest
National Computer Camp . . . 35
Magazines . . . 2
Model I/III/4
Programs
Alphabet by Michael Lofy . . . 27
Addition Drill/Drawing Pad by Adam Roth . . . 30
Fast Multiply by James Evans . . . 26
Maillist by Tom Kilbride . . . 24
Sound Generator by Jerry Stajduhar . . . 30
Word Counter by Joseph McCarthy . . . 32
TRS-80 Model 4 Desktop Computer by Bruce Elliott . . . 17
Model II/12/16
Repairing a Model II Boot Error by Curt Humphrey . . . 12
Pocket Computer
PC-2 Assembly Language-Part 3 by Bruce Elliott . . . 33
PC-2 Input Syntaxes by Peter Levy . . . 36
Programs
GMT Converter by Charles Blatt . . . 37
IRA Account by Harold Wright . . . 37
Resistors by Charles MacLeod . . . 38
X/Y Graph by David Baker . . . 37
View From the Seventh Floor by Jon Shirley . . . 4

Prices shown in TRS-80 MICROCOMPUTER NEWS are in U.S. Funds.



View from the 7th Floor

by Jon Shirley
Vice President
Radio Shack Computer Merchandising

THANKS To Mr. Bob Snapp of SNAPPWARE for his valuable help. Bob supplied us patches to solve the slow BASIC disk access under Model II TRSDOS 4.0 and for a problem with TRSDOS 2.0b using the thinline drives. All Bob asked for his efforts was official recognition and this is it.

SNAPPWARE produces enhancements to the disk BASIC for Model II and III for programmers. Bob has some very neat stuff and I speak as a user on my Model III. If you are interested, write to SNAPPWARE at 3719 Mantell, Cincinnati, Ohio 45236.

New TRS-80's just keep arriving like magic. First the Model 12, then the PC-4 pocket, then the Model 100, and this month you can go see the new Model 4. We think this new baby answers all the requests we have had for a low priced computer with a bigger screen and more memory. The Model 4 features an 80 by 24 screen, an enhanced keyboard with function keys, 64K expandable to 128K, double speed operation and even a built in speaker. It comes with an all new operating system that is loaded with features and the latest in BASIC's. It is also CP/M compatible and, as far as thousands of hours of testing can prove, it's 100% compatible with the Model III.

Put in a Model 4 disk and it's a Model 4, put in a Model III disk and it's a Model III. It also has a nice looking near white case which has also been requested from time to time. We do have an upgrade kit for the Model III owner that makes the III a full 4 except for the case color. It is $799 which is a lot, but it requires replacing the entire main circuit board and the keyboard. Since the Model 4 was so changed, even the plug in chips from a III cannot be used in a 4. One note, the Model 4 features are disk only. The cassette version operates only as a Model III, until it is upgraded to disk.

And we are not through introducing new computers, yet. Keep tuned in for more new products. This is an exciting industry where advancing technology will bring us all more personal computing power that we could have dreamed of just 5 years ago.

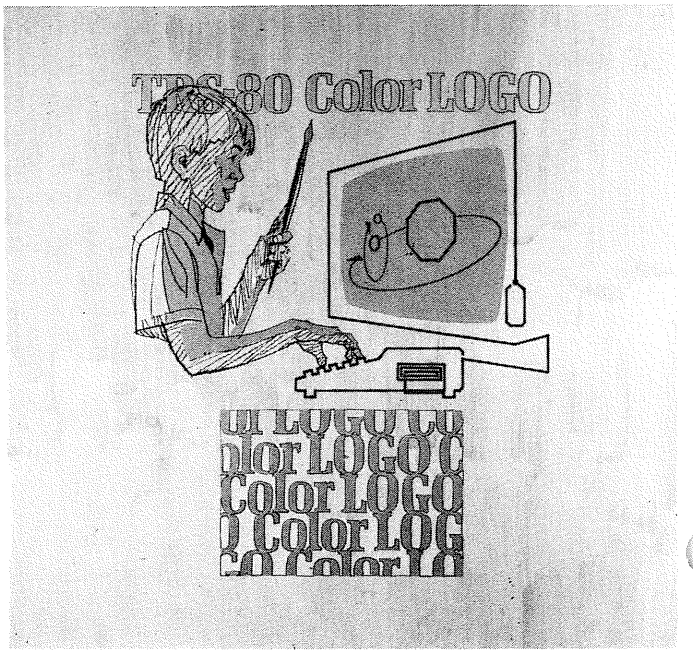
I would like to thank all of you who have applied to our software support program for software houses. We have been overwhelmed with the response. For you TRS-80 owners, this program will mean more availability of software for your TRS-80 from outside suppliers. By mid-February over 1,000 companies had signed up, and that means a lot of people are out there writing software for our computers. We are really proud of the vote of confidence and we will continue to expand the support department.

The subject of this issue is home use, and there is one home item that I think is really great for you Color Computer owners with small children. It's Color LOGO, now available as a Program Pack for any 16K Color Computer. LOGO is a very easy language to learn and lets the child produce graphics. The basic graphic element of LOGO is called a "turtle". The turtle is moved around the screen with simple commands to produce the graphics. Our LOGO has the unique ability to "hatch" multiple turtles which can be run by separate programs simultaneously.

Another neat LOGO feature called doodle can be used by children too young to read to move the turtle just by pressing keys. Each movement is saved in memory and can be replayed by pressing another key. Perhaps the best feature of our LOGO is that it's only $49.95 and does not require an expanded computer. That makes it by far the lowest cost LOGO available from anyone.

A lot of the surveys I read say that girls are not using home computers, or even TV games much. I hope that is not because parents assume that computers are for boys. My 12 year old girl spends hours at our Color Computer when she is not riding her horse. She plays games, tests new programs for us, uses Micro Painter, and has done a little programming. If your kids spend all their free hours at the boob tube, why not let them spend those hours on a color computer and learn at the same time? And that includes the girls. It's not just that I want to sell lots of computers, which I do, but that the world is getting computerized, and future employment for many of today's children will be in information processing, and that means computers. A little knowledge now might be very valuable later.

Until next month.



Radio Shack Dot-Addressable Printers

Why write a letter when you can express it? Dot-addressable printers, like the DMP series, can do much more than just print, so why not jazz it up a little? You can now transform printing into creative drawing, as the capabilities of dot-addressable graphics is limited only by your imagination.

A few of the new printer manuals have some programming examples using data statements, the most efficient way to manipulate large strings of data, because it saves time typing. Presented below are similar programs that should work on any Radio Shack dot-addressable printer.

Note: The output of the printer in bit-image mode depends on how the printer is configured (i.e. — is elongation on or off, etc.) If you want repeatable results, always set the printer to a known configuration.

Please key in the following program to see what happens.

Note: If the DMP printer has the rotary switch in correspondence mode, it will not accept graphic codes, so be sure your printer is set up for bit-image mode. This can be done at power-up using the switches, or by sending the proper software codes to the printer.

```
1Ø LPRINT CHR$(18);
2Ø READ N
     : IF N=999 THEN LPRINT CHR$(3Ø);
     : END
3Ø IF N=888 THEN LPRINT
     : GOTO 2Ø
4Ø LPRINT CHR$(128+N);
5Ø GOTO 2Ø
9Ø DATA 7, 25, 97, 1, 5, 61, 5, 1, 97, 25, 7, 888
91 DATA 56, 38, 33, 32, 47, 41, 41, 32, 33, 38, 56,
    999
```

Now try this . . .

```
4Ø IF N>=Ø THEN LPRINT CHR$(N+128);
     : GOTO 2Ø
5Ø READ NN
     : LPRINT CHR$(28); CHR$(-N); CHR$(NN+128);
6Ø GOTO 2Ø
9Ø DATA 124, -3, 2, -9, 113, -3, 98, 124, 12Ø, 112,
    888
91 DATA 119, -3, 28, 6Ø, 6Ø, -4, 56, -3, 57, -3, 1,
    127, 127, 124, 888
92 DATA 1, 2, 6, 14, 12, -8, 28, -3, 3Ø, 15, 15, 7,
    999
```

Line 50 implements the repeat feature. A negative number will indicate the number of times to repeat the following character string.

What about SuperSCRIPSIT? SuperSCRIPSIT allows 20 user defined printer code keys, and each key can store up to 11 printer codes or other data. That is 220 codes!

Why not utilize this feature with dot-addressable graphics. The first program 'draws' the Tandy logo. Let's try to recreate this in SuperSCRIPSIT . . .

In SuperSCRIPSIT, choose the System Setup option at the main menu, then choose C for printer codes. The Unit column is the width of the printable characters. Type in the following example, and stipulate the width as 4 on all three lines.

```
18 135 153 225 129 133 189 133 129 225 153
135 10 30 8 22 18 184 166 161 160 175
169 169 160 161 166 184 30 27 30
```

The number following the backspace (8 is the backspace, you may need to change the 22), may vary depending on your printer. Check your printer manual for the correct values. You could devise your own company logo (or maybe your signature in script) using a similar technique.

The following program prints old English letters. The letters to be printed must be uppercase alphabetic characters, all other characters will print as a blank space.

The program could be enhanced greatly, so I hope that this may have given you some new ideas.

```
1 CLS
     : CLEAR 1ØØØ
     : DEFINT N
     : DEFSTR A
     : DIM A(26, 2)
2 PRINT @ 47Ø, "One Moment Please...."
3 LPRINT CHR$(18)
1Ø FOR N1=1 TO 26
11 FOR N2=1 TO 2
12 READ N
     : IF N=999 THEN 15
13 A(N1, N2)=A(N1, N2)+CHR$(N)
14 GOTO 12
15 NEXT N2, N1
17 CLS
2Ø INPUT "Message"; AA
22 IF AA="QUIT" THEN LPRINT CHR$(3Ø);
     : END
24 FOR N1=1 TO 2
26 FOR N2=1 TO LEN(AA)
28 NN=ASC(MID$(AA, N2, 1))
3Ø IF NN<45 OR NN>9Ø THEN LPRINT STRING$(7, 128);
     : GOTO 4Ø
32 FOR N3=1 TO LEN(A(NN-64, N1))
34 LPRINT CHR$(128+ASC(MID$(A(NN-64, N1), N3, 1)));
36 NEXT N3
38 LPRINT CHR$(128); CHR$(128);
4Ø NEXT N2
42 LPRINT
     : NEXT N1
44 GOTO 2Ø
5Ø DATA Ø, Ø, Ø, 2, 1, 1Ø1, 57, 38, 63, 112, Ø, Ø, Ø,
    999
51 DATA 2, 1, 1, 1, 3, 2, Ø, Ø, Ø, 1, 3, 2, 1, 999
52 DATA 2, 1, 17, 126, Ø, 126, 9, 9, 14, 112, 999
53 DATA Ø, 2, 1, 1, 1, 2, 2, 2, 1, Ø, 999
54 DATA 16, 124, 2, 125, 1, 1, 2, 2, 64, 999
55 DATA Ø, Ø, 1, 2, 2, 2, 2, 1, Ø, 999
```

```
56 DATA Ø, 2, 18, 122, 5, 121, 1, 2, 124, 999
57 DATA 2, 1, 1, 1, 3, 2, 2, 1, Ø, 999
58 DATA 16, 124, 2, 125, 17, 17, 34, 2, 64, 999
59 DATA Ø, Ø, 1, 2, 2, 2, 2, 1, Ø, 999
6Ø DATA Ø, Ø, 2, 1, 17, 126, Ø, 126, 17, 2, 2, 1, 999
61 DATA 2, 1, 1, 1, 3, 2, 2, Ø, Ø, Ø, Ø, Ø, 999
62 DATA 16, 124, 2, 64, 62, 17, 1Ø, 1Ø, 113, 999
63 DATA Ø, Ø, 1, 4, 4, 4, 4, 1, 1, 999
64 DATA 2, 17, 125, 2, 126, 17, 9, 16, 96, 999
65 DATA Ø, 2, 1, 1, 2, 2, 33, 16, 15, 999
66 DATA 2, 1, 17, 126, Ø, 126, 1, 1, 2, 999
67 DATA 1, 2, 2, 1, Ø, 1, 2, 2, 1, 999
68 DATA Ø, 2, 1, 17, 126, 1, 126, 1, 999
69 DATA 2, 1, 1, 3, 2, 2, 1, Ø, 999
7Ø DATA 2, 17, 61, 2, 126, 17, 49, 5Ø, 8Ø, 999
71 DATA 2, 1, 1, 3, 2, 2, Ø, 1, 2, 999
72 DATA 2, 17, 61, 2, 126, 1, 1, 2, Ø, 999
73 DATA Ø, 2, 1, 1, 1, 3, 2, 2, 1, 999
74 DATA 2, 17, 126, 1, 17, 126, 18, 125, 1, 1, 126,
     Ø, Ø, 999
75 DATA Ø, 2, 1, Ø, 2, 1, Ø, 1, 2, Ø, 1, 2, 1, 999
76 DATA 2, 17, 127, 2, 125, 1, 1, 126, 17, Ø, 999
77 DATA Ø, 2, 1, Ø, 1, 2, Ø, 1, 2, 1, 999
78 DATA 16, 124, 2, 32, 3Ø, 1, 125, 2, 124, 999
79 DATA Ø, Ø, 1, 2, 2, 2, 2, 1, Ø, 999
8Ø DATA 2, 17, 126, 2, 125, 1, 1, 126, 999
81 DATA 2, 1, 63, 1, 2, 2, 2, 11, 999
82 DATA 16, 124, 2, 32, 3Ø, 1, 125, 2, 124, Ø, 999
83 DATA Ø, Ø, 1, 2, 2, 2, 2, 1, 2, 2, 999
84 DATA 2, 17, 126, 2, 125, 17, 17, 11Ø, Ø, Ø, 999
85 DATA Ø, 2, 1, Ø, 1, 2, Ø, 1, 2, 1, 999
86 DATA Ø, 22, 41, 41, 41, 42, 17, 96, 999
87 DATA 2, 1, 1, 1, 2, 2, 1, Ø, 999
88 DATA 2, 17, 121, 5, 1, 62, 2, 126, 1, 64, 999
89 DATA Ø, Ø, Ø, 1, 2, 2, 2, 2, 1, Ø, 999
9Ø DATA 17, 126, Ø, Ø, 126, 1, 126, 1, Ø, 999
91 DATA 2, 1, 2, 2, 2, 1, 3, 2, 1, 999
92 DATA 2, 17, 126, Ø, 126, Ø, 1, 126, 1, 999
93 DATA Ø, Ø, 1, 2, 2, 2, 1, 1, Ø, 999
94 DATA 2, 17, 126, Ø, Ø, 126, 17, 126, 16, Ø, 1,
     127, 999
95 DATA Ø, Ø, 1, 2, 2, 2, 1, 1, 2, 2, 1, Ø, 999
96 DATA 66, 1, 17, 18, 124, 18, 17, 1, 66, 999
97 DATA 1, 2, 2, 1, Ø, 1, 2, 2, 1, 999
98 DATA 2, 17, 126, Ø, Ø, 126, 1, 126, 1, 999
99 DATA 34, 17, 18, 34, 34, 33, 19, 12, Ø, 999
1ØØ DATA 2, 1, 65, 33, 17, 1Ø, 6, 34, 17, 16, 96, 999
1Ø1 DATA 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, Ø, 999
```

To center and print on mailing labels, make these changes. Spacing may differ slightly, depending on printer.

```
23 NX=9-(LEN(AA)/2)
27 LPRINT STRING$(NX*9, 128);
42 FOR N=1 TO 8
     : LPRINT
     : NEXT
```

Computer Customer Service Address and Phone Numbers

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

Model I/III Business Group (817) 390-3939
Model II/16 Business Group (817) 390-3935
Languages and Compilers (817) 390-3946
Color and Pocket Computer Group (817) 390-3944
Hardware and Communications Group . . . (817) 390-2140
Educational Software (817) 390-3302
Games, Books, and New Products (817) 390-2133
Newsletter Subscription Problems (817) 870-0407

Computer Clubs

CINTUG (CINcinnati TRS Users Group)
Lee Penn, President
6798 Weatheridge Drive
Cincinnati, OH 45230

COLUMBUS AND CENTRAL OHIO COLOR COMPUTER CLUB
c/o Stan Saunders
4984 Brewster Drive
Columbus, OH 43227
1-614-837-3672

FRONTIER MIDDLE SCHOOL COMPUTER CLUB
Prairie Street
Brookston, IN 47923

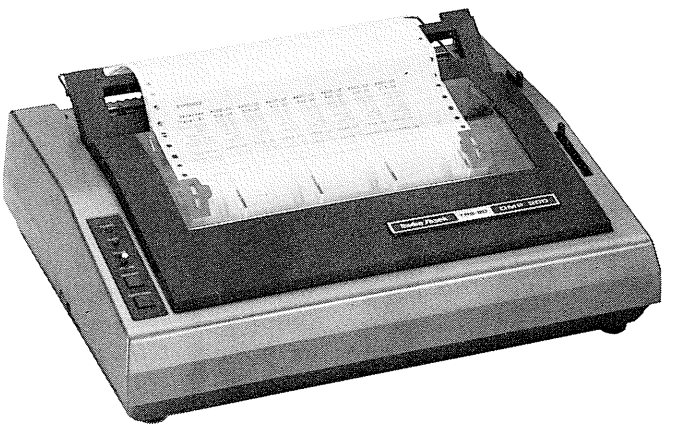
GULF COAST COMPUTER CLUB
c/o Jim Johnson
131 Fox Run
Port Richey, FL 33568
813-863-7954

MIDLAND COLOR COMPUTER CLUB
c/o Neil Drake
709 Coolidge
Midland, MI 48640
517-631-2939

PENSACOLA, FLORIDA USERS CLUB
TRS-80, Model III Users
Dr. Frank Biasco, President
9759 Pickwood Drive
Pensacola, FL 32504
904-477-1912

TCUG
P.O. Box 2826
Fairfax, VA 22031
703-820-6847 (Don Gruenther, Secretary)
703-836-0384 (TCUG BBS)

TWIN CITIES COLOR COMPUTER CLUB (TW3C)
c/o Paula Giese
515 West 25th Street
Minneapolis, MN



CompuServe Offers a Variety of Games and Entertainment Features

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

DICTIONARY OF COMPUSERVE GAMES AND ENTERTAINMENT FEATURES

The CompuServe Information Service is pleased to bring you a variety of games from Adventure to Wumpus.

ADVENTURE

The computer is your eyes and hands as you fight your way past snakes and dwarves in Colossal Cave in search of treasure.

BACKGAMMON

The computer is your backgammon board; play the computer or someone else.

BANSHI

A game for the explorer, leading you to forgotten territories where mere mortals have never been.

BLACKJACK

Play against the dealer with $500 to start your betting.

BRIDGE

Play bridge with the computer.

CHESS

Your chance to match wits with the computer.

CIVIL WAR

A simulation which lets you fight actual civil war battles.

CONCENTRATION

Like the TV games show, match prizes hidden behind numbered doors.

CRAPS

Big Lefty is your opponent as you roll the dice for points.

CUBE SOLVER

Explicit instructions on how to manipulate the colors on the frustrating cube puzzle.

DECWARS

Real-time space battle game for up to 10 players.

ELIZA

On-line counselor who helps you talk through your problems.

FANTASY

Battle the enemy in this role playing game similar to Dungeons and Dragons.

FASTERMIND

A logical guessing game where one player makes up a secret code, and the other player tries to guess it.

FOOTBALL

You call the plays: half-time and end-of-game statistics given.

FURS

You are a 17th Century French fur trapper.

GOLF

Play golf by selecting your clubs while a commentator explains the course as you play.

GO-MOKU

Line up the Xs and Os. This is similar to tic-tac-toe.

HAMMURABI

As the ruler of ancient Sumeria, you make decisions which determine the well-being of your people.

HANGMAN

A word guessing game. Guess in eight tries or be hung.

LUNAR LANDER

Land your ship safely by selecting the correct fuel rate.

MAZE

The computer creates a maze to challenge your skills.

MUGWUMP

You have 10 tries to find four Mugwumps hidden on the grid.

NEW ADVENTURE

An expanded version of the exciting Adventure game.

OTHELLO

Capture the computer's "markers" on an 8×8 board.

REAL-TIME TREK

A real-time version of Star Trek with multiple players.

ROULETTE

Bet on roulette with the computer as the wheel.

SCOTT ADAMS ADVENTURE

Nine new adventure games including Pirates Adventure, Voodoo Castle and Pyramid of Doom.

SCRAMBLE

Unscramble the scrambled word before your points run out.

SPACE WAR

Space age game for up to eight players.

STAR TREK

Use warp engines to power through the 64 galaxy quadrants and fight the enemy.

WUMPUS

Get your bow and arrows and hunt through a series of caves for the Wumpus.

BE THE STAY-AT-HOME TYPE WITH COMPUSERVE

The CompuServe Information Service provides home uses which are perfect for nearly every member of your family. Even the hard to please members like your mathematicalized son Bobby or your budget-minded husband Harold. You could almost say that it's possible to hibernate with CompuServe. You wouldn't have to leave your warm, cosy, comfortable . . . well you get the point. You could plant yourself or your family in your living room at the terminal for days on end if you wanted. So, what do we have that's so special and that can keep you or those "difficult ones" occupied for hours?

AIN'T WE GOT FUN

We have games and games and more games. We also have entertainment features. We have games for the adventurer. If you love to stay at home but hate to miss anything, we have just the things for you. Adventure, and Scott Adams' other Adventure games will let you wander and explore without ever leaving your armchair. The original Adventure game lets you be in command. You say what part of the cave to explore, which treasures you want, and how to get past all the obstacles. A perfect score of 350 points will make you master of the original adventure, and ready to head on to the New Adventure. This is the enhanced version which provides more challenges and more dangers ahead for the skilled adventurer. A score of 751 makes you conqueror of the Centipede's Lair.

Scott Adams provides nine adventure games with varying levels of difficulty. Start at the beginning and work up, or start at the top and work down. Either way, you'll find plenty of mystery and challenge with Adventureland, Pirates Adventure, Mission Impossible, Voodoo Castle, The Count, Strange Odyssey, Mystery Funhouse, Pyramid of Doom, and Ghost Town.

If you're more interested in being a space cadet than a Sinbad then CompuServe's outer space is your outlet. We offer DecWars, and its offspring: MegaWars, Star Trek, Luner Lander, Pirates, and Space War. DecWars or MegaWars offers a real time space battle for between one and 10 people. Star Trek lets you go where no man has gone before by commanding the Enterprise. Now, you get to phaser blast, torpedo, and destroy as many Klingon vessels as you can in a given amount of time.

The Luner Lander helps you take one small step for your household, one giant step for you by landing the module on the moon. By telling the computer what to do, you decide whether you will land safely or perish above the moon's surface.

Quite the opposite of Luner Lander, is Space War. Rather than landing, you're taking off into space to seek out and destroy other ships in your galaxy. This is a real time battle, so you could be playing with one other person or eight.

One other space game is Pirates. This is a real-time Star Trek type game. Out there in space you're battling against the computer and the clock to come out ahead of your enemy.

Other games and entertainment include an astrology and biorythms calculator, psychiatrist/patient simulator, sports games, chess, blackjack, craps, and a whole variety of mind involving games. All members of the family can find something just for them, including the mathematician, and the budget director. Just check the listing at the beginning of the game section on CompuServe for suggested age, classification, number of players, special requirements, screen width and direct access page.

OH, I JUST LOVE A SALE!

Do you know of someone who loves to shop? Someone who loves to spend money? Someone who has a personal computer? You can combine all three of these characteristics without leaving your home or depleting your checkbook or VISA credit. How? Via CompuServe shopping services. We have shopping services offering a wide range of merchandise, plus a variety of prices and discounts.

Yes, now you can have that full stereo system that Harold wouldn't let you buy in the department store because it costs too much. As you sit in your easy chair, do some easy shopping with Comp-U-Store. As a member, you have your choice of over 50,000 brand name products all available for home delivery, and not at the department store price you quoted Harold.

Membership fees are $25 for one year, and $40 for two years. This membership includes a free VISA card that is good anywhere. Comp-U-Store offers appliances, cameras, flatware, and video equipment from names such as Admiral, Canon, Omega and Sony. All items are sold at 10 - 40 percent discounts with full guarantees and home delivery. Comp-U-Store also offers exceptional bargains from their already discounted prices, and the ability to buy something really nice for Harold and have it shipped directly to him as a gift at no extra cost.

If Harold is a fan of National Public Radio, you can use CompuServe's NPR Cassette Gazette. NPR is the only national noncommercial satellite radio network. Its over 265 member stations provide award-winning educational, cultural, news, and information programming. Some of the top-

ics NPR cassettes cover include "The Computer as the Next Revolution," "Ragtime Women," "Sputnik and Beyond," and something to especially perk Harold up, the "Comedy Journal."

If Harold would rather be exercising, CompuServe can switch him over to The Athlete's Outfitter, the 60-year-old sporting goods store with an extensive line of sports clothing, equipment, and the utilities that go along, or shall we say under, the clothing worn while using the equipment. Athlete's Outfitter lets you browse, and then order from our catalog. Just get to what you would like to order, read the description, colors and sizing, and then follow the prompts for ordering. You can even figure out your sales tax before you leave the "Outfitter," making your purchase as smooth as a ski run with new snow and waxed skis.

Of course, if Harold wants to be a well rounded person, he must have some music mixed in with his exercise and his cultural cassette library. CompuServe's Music Information Service can offer him up-to-date information on all forms of music and music information. Shoemaker's Music Inc., out of Florida, provides answers to questions, retail music business operation, or music theory. The Feature section offers articles on woodwind care, or music synthesis, or any other interesting topics provided by interested readers . . . like Harold. Music Information also offers an Ask the Experts section, an Interviews section and the Top 40 in rock 'n' roll, country, and other medias and fields. And, if you want to purchase, they even provide you with a mall. You can browse through new items added weekly. This is the place to be for accessories, instruments, and any music in print. And of course, if they don't have it, they'll try and get it or help you find another supplier.

ALL IN THE FAMILY

How do you answer when people say, "How's the family, Harold?" Do you say "Oh, fine"? Or, does your answer specifically reflect tensions you have concerning your family life? Do you answer with specifics such as "Well, Joe comes home late at night, Sally has stopped doing her homework and is skipping classes, and Jean decided to quit her job and visit her mother this week." If you do, you're probably in company with a great number of other folks who are confronting the everyday task of raising a family and being a part of one at the same time.

CompuServe offers dataFAMILIAE and dataFAMILIAE's Special Interest Group (SIG). In a time when families and society's roles are changing and shifting, it's hard to know which direction to take to get the best results. dataFAMILIAE provides quick, easily read, accurate information, and knowledge and training for the fundamentals of parenting and family living. dataFAMILIAE attempts to focus on the "wholeness" of it all, while helping you to gain confidence in handling familial tasks.

dataFAMILIAE provides five sections of everyday living material including: Functioning Mentally, Functioning Physically, Nurturing Moral Development, Fitting Together as a Family, and Special Matters of the Family. These sections are geared toward helping the individual with his/her growth, preparing and participating in a family and bringing all of our most positive aspects into a family setting.

In conjunction with the dataFAMILIAE service, the dataFAMILIAE Special Interest Group is also dedicated to parenting and family life. Users of this SIG can interact on matters such as mental, physical and moral growth in the family, individual roles, communication, decision-making, household management, nurturant experiences, dysfunction plus others.

The SIG is managed by Dr. Bob Lindamood and his associates, who operate a management consulting organization in Columbus, Ohio. Bob and his associates have qualified backgrounds in child development and family relations in order to provide you with whatever answers you may need.

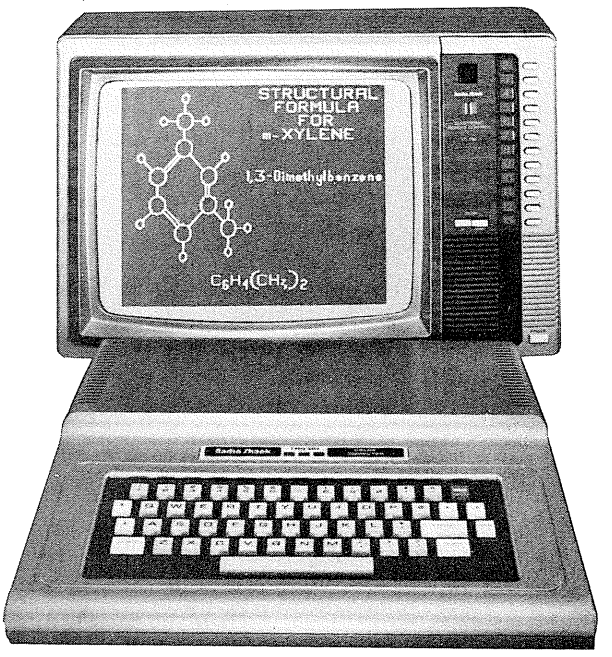
Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, Editorial Director, or Jacqueline A. Farthing, Assistant Editor, CompuServe Information Service, 5000 Arlington Centre Boulevard, P. O. Box 20212, Columbus, Ohio 43220 or through Feedback, main menu item 5, CompuServe User Information.

Educational Color Computer Users Network

Dr. Paul Kimmelman, of Norton City Schools in Norton, Ohio, is establishing a "network" of educational users of Radio Shack's Color Computer. Dr. Kimmelman is requesting the following information from educational users of the Color Computer:

1. Computer coordinator's name
2. School district
3. Address
4. Educational uses of Color Computers
5. Software being used and evaluations of it
6. Any additional information you might have

This information should be sent to:
Dr. Paul Kimmelman
Assistant Superintendent
Norton City Schools
4128 S. Cleveland-Massillon Road
Norton, Ohio 44203



COMMUNICATIONS CORNER

Al and Dru Simon
P.O. Box 2910
Fort Worth, TX 76113-2910

Hi and welcome back to Communications Corner. Thank you for your comments and questions, we appreciate them highly. Please keep them coming!

For the next few columns we are going to delve fairly deeply into the mysteries of RS232C; that oft-mentioned and vaguely understood convention which everyone knows is good to have—though few understand why. At last we are going to tell you EVERYTHING YOU ALWAYS WANTED TO KNOW ABOUT RS232C BUT WERE AFRAID TO ASK!

First the term itself. RS232 refers to a specific Electronics Industries Association (or EIA) standard which defines (rather widely) one method for interfacing two or more pieces of data equipment with each other. Currently RS232C interface is by far the most widely used. Frequently, peripherals such as printers, voice recognition devices, modems, terminals (in the case of the Mod 16) or even whole computers (in the case of the Network 3) are controlled through this port.

As a part of this series on the RS232C we will be discussing different types of serial communication, PINS and what they are used for, and this month we will start discussing the type we're interested in and how that type (known as Bit Serial Asynchronous or Asynchronous Bit Serial) works.

We will start our discussion by describing exactly what happens over the telephone line during the transmission of data. In this process your computer may be set to either Transmit data or to Receive data. There are four tones involved in data transmission; and their frequencies are associated with each transmission or receive. In other words, each specific frequency means something in particular to your computer.

The Transmit frequencies are as follows:

To originate a MARK, frequency is 1270 HZ
To originate a SPACE, frequency is 1070 HZ
To answer a MARK, the frequency is 2225 HZ
To answer a SPACE, the frequency is 2025 HZ

Notice that they are only separated by 200HZ. In Receive, the frequencies are organized in pretty much the same way, but in reverse. Specifically:

To originate a MARK, frequency is 2225 HZ
To originate a SPACE, frequency is 2025 HZ
To answer a MARK, the frequency is 1270 HZ
To answer a SPACE, the frequency is 1070 HZ

The switching of these frequencies tells your modem whether a 0 or a 1 (also known as a SPACE and MARK respectively) is being transmitted or received. Simply put, the job of the modem is to turn frequencies received over the phone line into binary 0s and 1s.

What your modem is looking for when you are communicating is the change between MARK and SPACE. The duration of this tone is controlled by your baud rate generator in your computer. The baud rate is simply the speed at which asynchronous data is transmitted and is the mathematical inverse of the time duration of the shortest signal element.

Let's now describe a data character. If you look at figure 1, you will see that there is a START BIT followed by a character, followed by one or more STOP BITS. (There may also be a parity bit included as well, but not necessarily. This is an option of YOURS—that is, when you set your parameters before calling another computer you are given the option to set your parity yourself.) The word or character length varies from 5 to 8 bits. Remember, that computers speak in binary, and each signal has its meaning. Binary 1 is considered to be a MARK, and binary 0 is considered to be SPACE.

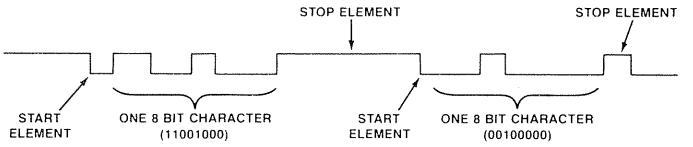


Figure 1.

The computer knows that a character is being sent when it recognizes the START BIT. It then reads the data it is sent until it receives a STOP BIT. Then it evaluates the information it has just received. The START element is a logic 0, and the STOP element is a logic 1. The STOP element is held until the START element of the next character is transmitted. There is no limit to how long your computer can wait for the next character; however, there usually is a lower-end time limit, and that depends on the system you're on.

With asynchronous communications (the type we're dealing with), there is no clocking or synchronizing or clock signal sent by the receiving computer to coordinate the information being sent across the line. (In synchronous situations, time intervals are used instead of a search for start and stop bits.)

One of the convenient features of asynchronous communication is the fact that transmission over long distances can be accomplished using a minimum of three wires twisted. These wires go to the PINS which we have described below.

Let's define these the functions of PINS. RS232C requires a 25 PIN connector called a DB25. The signals we are most interested in are transmitted through these PINS as listed below:

Pin 1—Hardware Ground
Pin 2—Transmit data
Pin 3—Receive data
Pin 4—Request to send data
Pin 5—Clear to send data
Pin 6—Data Set ready
Pin 7—Signal Ground
Pin 8—Carrier Detect

Pin 20—Data Terminal Ready
Pin 22—Ring Indicator (used in some of the newer Auto Answering modems)

In numerical order what these PINS do is this:

PIN 1—CHASSIS GROUND. This is a protective ground and should always be in installed. If you are using Radio Shack equipment this has already been done for you. If you are designing your own equipment or "home-brewing" you must make sure that PIN 1 goes to chassis ground on your computer.

PIN 2—TRANSMIT DATA, or TD. Words (or the entire figure 1 including the STOP BIT, START BIT, parity bit, and character which is called a DATA WORD) are transmitted to the computer you are talking to along this line. This will be held in MARKING condition (or binary state of 1) during intervals when no data is being transmitted.

PIN 3—RECEIVE DATA, or RD is the line that you will receive data words on, and that signal is also going to be held in a MARK condition at all times when no characters are present.

PIN 4—REQUEST TO SEND, or RTS is used to control the direction of communications. When in full duplex, the SPACE condition holds the data device in transmit, the MARK in a non-output mode. When in half Duplex channel, the SPACE condition holds the communication equipment in a Transmit mode and conversely the MARK condition maintains the equipment in a Receive mode.

PIN 5—CLEAR TO SEND DATA, or CTS. This is a signal which your computer is expecting to receive (whether generated by the modem or software). It indicates to the device you are communicating with the fact that the modem is ready to transmit. The SPACE condition tells the equipment that the modem is ready to accept data. The MARK condition tells it that it is NOT ready to receive data.

PIN 6—DATA SET READY, or DSR. This signal indicates to the equipment you are talking to that your equipment is prepared to accept data. A SPACE condition signifies that all is ok, and a MARK condition tells it that it is NOT ready.

PIN 7—SIGNAL GROUND. This is used to insure data integrity in respect to transmission interference and is used by the system as a "0 reference voltage."

PIN 8—CARRIER DETECT, or CD. If this is in the SPACE condition, it signifies to your computer that a carrier from another device is present. Sometimes this line is tied off with Pins 20 and 6 to force a certain condition (we'll explore that later). If this line is in a MARK condition, no communications can proceed.

PIN 20—DATA TERMINAL READY, or DTR. This is similar to the carrier detect, but it tells the communications equipment that you are speaking to that your hardware is ready to receive information.

PIN 22—RING INDICATOR, or RI. This goes to an ON or a SPACE condition every time the phone rings. Whether or not your computer recognizes this signal depends on your modem. This is true for most signals, with the possible exception of PIN 7 (signal ground) and PINS 2 and 3.

It is frequently enough to have PINS 7, 2, and 3 present in a DB25 connector to insure that you will be able to speak to other machines. (This is what we meant by "3 wires twisted.") In this type of situation PINS 6, 8, and 20 should be tied to each other at both ends of the cable. In effect, what this results in is a situation where you are getting the CLEAR TO SEND, CARRIER DETECT, and DATA TERMINAL READY (the three hardware checking signals) all tied together in a positive condition which will allow you to communicate.

All this is what happens electronically through your RS232C during Asynchronous Bit Serial communication. Your terminal program takes care of all these functions for you automatically.

RS232C also handles many types of Synchronous communication, plus other types of Asynchronous communication, much of which we will be discussing in future columns.

As an additional feature to our column, each month we will try to answer some of the mail we have received from you. We are pleased by the response we have already had, and look forward to more letters. Let's talk!

Kelman Aiken
Stateline, NV

I've heard several references to bulletin boards, but have never tried one. My hesitation has been that all appear to be aimed at the Model Is and Model IIIs. I have a Model II.

I hope your upcoming articles will describe how to sign on a Model II, (from TRSDOS Ready) and how to operate the printer for downloading and uploading.

Dear Mr. Aiken,

Thanks very much for your letter. While we have planned on doing a more complete article on Model II, we will briefly tell you how to get "up" to get you started and will in fact cover Model II communications (and Model 16) in the future. The bulletin boards are not so much oriented for the Models I and III, but more oriented to a particular communications protocol which the Models I and III use. You, on the other hand have to do everything manually on your Model II. What follows below is EXACTLY what you have to do to get onto a Bulletin Board using 300 baud, 8 bit word, no parity, and 1 stop bit. (These are the most common settings; however, 300, 7, Even Parity, and 1 stop bit are also commonly used. If one doesn't work, try the other.) What you see below between the asterisks is what you must type in response to the TRSDOS prompts we have illustrated.

```
TRSDOS READY
*SETCOM A=OFF*
PORT A IS DISABLED
TRSDOS READY
*SETCOM A=(300,8,N,1)*
PORT A IS ENABLED AS 300 BAUD 8 BIT WORD NO
    PARITY 1 STOP BIT
PORT B IS DISABLED
TRSDOS READY
*TERMINAL*
```

At this point you will be presented with a menu. On that menu you will see "V" for video filter. Toggle that option ON by pressing V. You will also see "L" for Auto Line Feed. Toggle this ON as well by pressing L. You are now prepared to "talk" to most bulletin boards. At this point you may either dial and connect your modem in the ORIGINATE mode, or activate your autodialer, if you have one. As soon as you have made the connection to your chosen Bulletin Board, you will see their opening message and can take it from there, following whatever instructions are printed on your screen for you.

In order to download, you must first return to your terminal menu by pressing the ⟨**BREAK**⟩ key. (You may return to this menu at any time during your connection to the Bulletin

Board by pressing the ⟨BREAK⟩ key. This will NOT disconnect you!) Press R from the terminal menu. This will open a "BUFFER" or storage place for the program you are about to take from the Bulletin Board. You will then be asked if you wish to reset your buffer. Answer with a YES. Then press T for Terminal, and you will be returned to your exit point on the Bulletin Board you are talking with. I suggest you open your buffer BEFORE you select the program you wish to download. Normally, the computer you're talking to will send signals to open and close your buffer for you, but with the terminal program on the Model II these signals are ignored, and you must manually open and close your buffer. Once you have received the program from the Bulletin Board, press ⟨BREAK⟩ to get back to your menu, then R to close your buffer (NOTE: all these commands are toggle commands . . . push once for ON and once for OFF). You then have an option to either dump the downloaded program to screen, printer, or disk. I do NOT recommend running the printer simultaneously with communications. I DO recommend that you dump your buffer to disk using the "C" command. You will be prompted for a filespec, at which point you may enter the name you wish to give the file. REMEMBER: The Bulletin Board CANNOT save files to your disk for you! You must do this yourself.

DO NOT toggle open your buffer before you have saved the program you have downloaded. If you do, DEFINITELY do not reset the buffer, or you will lose your program!

Now, we have downloaded a file from the Bulletin Board but it has a header of all sorts of garbage which came before your program, such as your selection of the program, and other things that might have been on the screen when you opened your buffer in the first place. I recommend that you use your SCRIPSIT disk to edit out these unnecessary characters and signals. Bear in mind that you have saved an ASCII file, and it must be converted by SCRIPSIT before SCRIPSIT can use it. It should be converted back afterwards as well. Please note: If any line in your ASCII file is longer than 240 characters, when trying to load it, you will get a DIRECT STATEMENT IN FILE error in BASIC. Use your SCRIPSIT to break the lines in half, but remember to put an appropriate line number in front of it!

Hope this helps you to get going online, and I hope you enjoy your sojourn through the Bulletin Board world!

Eric Greene
W. Palm Beach, FL

As headquarters system for the GREENE MACHINE; a TRS-80 based network of Bulletin Boards, and in consideration of the number of systems I have across the country, I was wondering why my network was not mentioned in January's column.

I was very pleased to see that your column promoted the use and knowledge of Bulletin Boards in general, and that is why I was hopeful to see my own listed with those you did mention . . . Perhaps in some future column you might correct this oversight.

Dear Mr. Greene,

I apologize for not having mentioned your system, but the list that was published was representative only and by FAR not meant to be a complete list.

For you and Mr. Joe Simon of Bullet-80 (who also communicated the same request) please note that I have listed below both your headquarters.

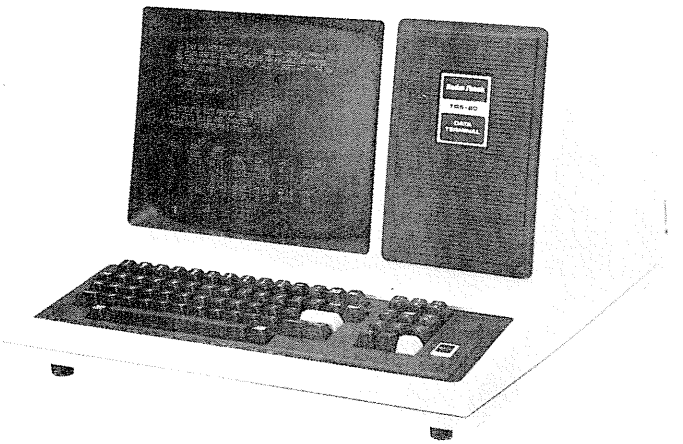
To all the other myriad Bulletin Boards and networks we have inadvertently missed mentioning, our apologies, but to publish a complete list would be impossible.

THE GREENE MACHINE - HQ
WEST PALM BEACH, FL
305-965-4388

BULLET-80 - HQ
DANBURY, CT
203-744-4644

MOUSE NET - HQ
BURLINGTON, VT
802-862-7023

Please watch for next month's column, in which we will continue our discussion of RS232C communications, as well as modem types and usages.



Repairing a Model II Boot Error

Curt Humphrey
921 Lurline Dr.
Foster City, CA 94404

As a Model II owner, I wish to report my discovery of a non-Radio Shack product which fixes the only serious problem I have experienced with my system in two years of heavy usage. It is my hope that other Model II owners can benefit from my discovery.

The product is called a SEEBEE and is sold by Systems Enhancement Engineering, P.O.Box 40215, Indianapolis, IN 46240. Its function is to eliminate the damage that will inevitably occur to a disk in drive 0 when the system is accidentally booted with the external drives turned off. Its use allows complete freedom to operate with the outboard drives off from the start, or to turn them off—or on—subsequently. Besides eliminating the nagging worry, it saves wear and tear on the outboard drives by allowing them to be shut down when not needed. Obviously, energy is also saved. The unit is attached to the computer externally, in-line with the ribbon cable to the external drives, and is completely out of view under normal conditions. No external power is required.

I have used the device for over a month and have found no way to defeat it. I would certainly recommend it for all Model II owners.

Editor's note: We are passing this information on to you, but you should be aware that it has not been tested by Radio Shack.

A Grading Technique Using SCRIPSIT

Dr. Timothy Little
Associate Professor
Michigan State University
College of Education

A DILEMMA FOR TEACHERS

"I've got three sets of papers to grade this weekend!" A thought like that can be enough to chill the heart of the most dedicated teacher. At the same time, there are few more important aspects of a teacher's responsibility than that of providing prompt, meaningful feedback to students regarding the quality of their work.

As a teacher for some twenty years in the public schools and at the university level, I must confess that meeting the "prompt feedback" responsibility has proven to be one of the most demanding and draining aspects of my work week. The problem is not so acute or pronounced with student work which lends itself to quantitative assessment. The prospect of grading an objective test which can be scored with an answer key and discussed in class item by item rarely provokes anxiety in the heart of the conscientious teacher. Furthermore, with the advent of microcomputers, this type of evaluation can be handled more quickly than ever before. With my TRS-80 Model III, Line Printer VIII, and some "homebrew" grading programs, I have been ripping through such sets of student papers with gusto and little fatigue for about two years now.

Unfortunately, student work which is written in essay form is another matter. Traditionally, such work is read by the teacher, who then assesses the quality of the student effort and annotates the paper with his or her remarks as to the student's performance. Since I am a firm believer in the vital importance of student essay work in the process of education, it is not unknown for my comments on papers to rival those of the pupils in length. But vital though it be, the spectre of a weekend of such grading has been enough to drive a teacher to drink, video games, or both. Specifically, there appear to be four aspects of grading student essays which contribute heavily to teacher antipathy toward this type of student assignment. Grading essays has often tended to be:

1. Disproportionately time consuming to a degree that can only be appreciated by practicing teachers.
2. Tedious and repetitious to the extent that an instructor checks each paper for the same core ideas as well as writing mechanics.
3. Cumulatively very tiring with the concomitant danger that the quality and length of teacher annotations decline from the first paper to the last.
4. Frustrating to the instructor because when student papers are returned to their owners, teacher comments will be lost to the teacher barring a massive and cumbersome copying operation.

Given the above, it is a wonder, and much to the credit of teachers in general, that essay assignments are given as frequently as they are in our schools.

SCRIPSIT AS AN ANNOTATING DEVICE IN GRADING

Introduction to the potential of SCRIPSIT as an aid to grading student essay work began inauspiciously. Faced with a self-imposed deadline for the return of a set of student papers, I was hurrying to annotate them and was railing at the fact that I had to repeat the same comments regarding writing mechanics on more than a few of the essays. At that moment, the proverbial lightbulb switched on. Before thirty minutes had elapsed, I had warmed up my word processing program and typed a series of the most repeated grammar-related comments, both laudatory and cautionary, into my machine. Set off in blocks and designated by code letters, they constituted a primitive menu of reactive comments. Using the menu required no more effort than that of typing in the pupil names and the applicable block letters for each student. Printed on fanfold paper, the comments were then cut out and stapled to each of the corresponding student papers. Inasmuch as I had saved the comments on cassette tape as well, I was also now in the possession of a complete set of comments for the class in the area of writing mechanics.

Alerted to the potential of SCRIPSIT for the rapid annotation of student papers, by the time the next writing assignment came due I had prepared a considerably upscaled version of my word processing scheme. Specifically, I had prepared a model answer to the question which I posed in the student essay assignment. For each of the major three or four points which I hoped/expected students to treat in their responses, I had developed a set of reactive statements of my own. These reactive statements were keyed to the various directions which student answers might be expected to take regarding the central points which I had identified. Each of these statements was then assigned a block designation. (My version of Model III SCRIPSIT 1.0, permits up to 23 separate block designations.) I also retained the grammatical reactive comments from my first experiment and assigned them the block designations which remained after all of my other reactive comments were covered.

I printed out a copy of my new "response menu," propped it against a typing stand, and set about the business of grading and "responding" to my student essays. Equipped with such a menu, I was able to create a formal essay response to each student's work by combining appropriate blocks, printing each set of blocks, and affixing same to

the student essays as before. An outline example of this procedure follows in Figure 1.

Figure 1. Instructor Response Menu

Type Student Name: _____

Major Points Students are Expected to Include in their Essays	Alternative Instructor Responses to Anticipated Student Answers		
Major Point #1	Block Statement	A B C D	
Major Point #2		E G I J	(F and H are reserved in SCRIPSIT 1.0)
Major Point #3		K L M N	
Grammar/Mechanics		O Q R S T U	(P is reserved)

Applied to the work of a specific pupil, the response menu allows me to create a personalized response for that essay almost as quickly as I can think. All that is required is for the instructor to type in the student's name and then the appropriate block letters. With the depression of the (ENTER) key, the word processor takes over and creates an extended paragraph of personalized responses keyed to the essay in question. Relieved of the drudgery of penning repetitious comments, a teacher may be encouraged by this procedure to further individualize his or her response to a student's work. This can be readily accomplished by typing in comments unique to a given paper as an add-on section to the block-generated prose. An outline example of the ease and simplicity with which this procedure might be applied to a hypothetical student paper follows in Figure 2.

Figure 2.

Type Student Name: John Smith _____

Major Points Students are Expected to Include in Their Essays:		
Major Point #1	Insert Block	B
Major Point #2		C E
Major Point #3		I M
Grammar/Mechanics		T U

Individualized teacher comments unique to John Smith's paper typed in addition to block inserts.

Print, cut out, and staple teacher's comments to John Smith's essay.

DIGEST OF THE ADVANTAGES OF USING SCRIPSIT IN THE ANNOTATION OF STUDENT ESSAYS

Capturing the range of payoffs inherent in the SCRIPSIT annotating technique described requires that we return to the teacher cost of essay grading described in the introduction to this article. By way of review, essay grading was characterized as frequently:

1. "Disproportionately time consuming. . ."

Clearly, use of SCRIPSIT can markedly reduce the amount of time that a teacher must spend annotating a given set of student essays. It was not uncommon for me to spend 3 to 5 hours per set of papers prior to my discovery of SCRIPSIT's potential. I would conservatively estimate that use of the SCRIPSIT technique has cut that real time by one half.

2. "Tedious and Repetitious. . ."

When annotating student papers there is a strong tendency for the mind to race ahead of one's ability to put thoughts to paper. That is, there is a "writer's lag" which occurs when the teacher's thoughts about a piece of student writing are dammed up behind the inability of his or her fingers to move quickly enough. Nowhere does this cause more frustration to an instructor than when the necessity of recording repetitious but crucial ideas on a student paper slows down your ability to react to truly unique features of that essay. Once again, SCRIPSIT can ride to the rescue of teacher creativity! My experiments suggest that the tedium factor all but evaporates with the ability to string together, almost instantaneously, appropriate blocks of comment; leaving the instructor fresh to append those unique ideas heretofore dammed up by "writer's lag."

3. "Cumulatively very tiring teacher annotations decline in quality and length from the first paper to the last."

A clear and present danger to the principle of equal treatment of student work is implied above. The flagging teacher who is still at it during the 11:00 p.m. news is not so likely to approach the 29th student essay with the same degree of intellectual fervor which marked his or her attack on paper number one, three hours earlier. Conscientious teachers are well aware of this problem, and employ a wide range of strategies to compensate for it, such as annotating papers in subsets of 5 to 10, with breaks scheduled between each batch. The SCRIPSIT annotation system should be viewed as a first-rate addition to the teacher's arsenal of compensatory strategies. Although I am happy to report that I emerge from my essay-grading sessions these days refreshed rather than numbed, I would also like to believe that, of equal importance, the principle of fairness in grading is well served by my use of SCRIPSIT. This because I believe that with word processing to aid me in the annotation process I am less likely to subliminally fault papers which have the misfortune to be located near the bottom of my grading pile. I might add in passing that the time saved in annotating as well as the sense

of efficiency associated with working within the SCRIPSIT system can be a morale booster of no small measure for the veteran teacher.

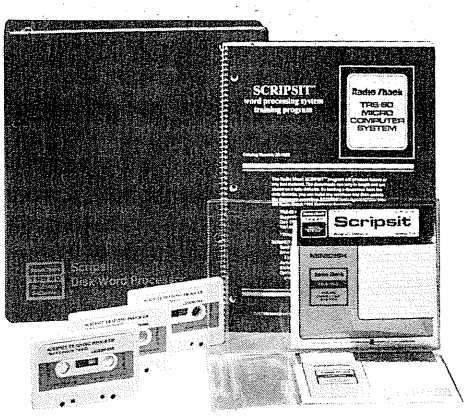
4. "Frustrating to the instructor because when student papers are returned...the teacher's comments will be lost . . . "

Clearly, the SCRIPSIT annotating technique solves this problem. By typing all of his or her responses to student papers on assignment "X" into the computer as a single batch of data, the teacher has the capability of saving the full set of annotations for that assignment on tape or disk. The ability to call up a full set of comments to screen or printer in itself represents a most valuable resource to the teacher who wishes to obtain a quick but crude profile of how the class as a whole fared on assignment "X".

There is, however, another way in which SCRIPSIT can be utilized to manipulate teacher annotation data which comes as an unexpected bonus of its editing function. SCRIPSIT, like other word processing packages, has the capability to carry out a "global search" across a manuscript for individual words within the text. That is, you specify a keyword that you wish to isolate, and the program will scan the text and identify all instances of that word. By identifying keywords for his or her reactive comments to students, a teacher may adapt this editing capability to quite a different use. By way of example: if the teacher wishes to identify all of the students who made error "y" in their essays and got teacher response "c" in return, he or she first loads the cumulative reactive comments for that essay assignment from tape or disk.The teacher then calls up a "global search" for all incidences of the keyword associated with block "c." Similarly, if the instructor wishes to tease out the names of all students who have a history of spelling problems in past essays, he or she has only to load past essay commentaries and institute a "global search" for the keyword associated with poor spelling. The potential uses of the "global search" function of SCRIPSIT seem limited only by the imagination.

SUMMARY

Clearly, the unorthodox uses for SCRIPSIT which I am proposing are primitive in their level of development. But, primitive as they are, they have already proved immensely useful to me in my efforts to be a better teacher; I can only hope that they prove equally useful to the readers of this article. More importantly, these ideas should serve to underscore the fact that there are potential uses of word processing packages for teachers that go far beyond that of the typing lab or typing form letters home to parents. Let's find them!



Creating a Network 2 Color Logo Lab

With Color **LOGO** for the TRS-80 Color Computer and the **Network 2 Controller**, you can set up a Network 2 Color LOGO lab in your classroom.

Color LOGO is Radio Shack's computer language for young learners. It permits students to control the "Turtle" on the screen using graphics commands. Students can also create routines called "procedures," which cause the Turtle to draw more complex graphics on the Color Computer screen.

Radio Shack's Education Division offers Color LOGO in two forms: on a diskette (for use with a 32K TRS-80 Color Computer with Disk Extended Color BASIC) and on a Program Pak (for use with any 16K or 32K TRS-80 Color Computer).

THE SETUP

The teacher's Host computer is a 32K Color Computer with a single disk drive. Disk LOGO is running in the Host computer.

A Network 2 controller connects the Host computer with each of the Student Stations. The "Baud Rate" switch is set at 1500, and the "Mode" switch is set at MPLX. Up to 16 Student Stations can be connected to the Host, via the Network 2.

Each Student Station is a 16K Color Computer with standard Color BASIC (the least expensive Color Computer). Each Student Station contains a LOGO Program Pak.

SENDING PROCEDURES TO THE STUDENT STATIONS

Perhaps the teacher has written a LOGO procedure, saved as procedure "A" on a diskette, to show the students.

At the Host computer, the teacher types,

(BREAK) (L) (A) (ENTER)

(to load program "A" from the Host disk)

Each student at a Student Station types,

(BREAK) (L) (T) (ENTER)

(to prepare the Student Station to receive the program A through the Network 2)

At the Host computer, the teacher types,

(S) (T) (ENTER)

(to send the procedure through the Network 2)

The LOGO procedure that was loaded from the teacher's Host disk, is now transferred to each Student Station. Each student can edit and run the procedure at his or her Station.

RECEIVING PROCEDURES FROM STUDENT STATIONS

The teacher may want to save on a diskette, a procedure a student has written at a Student Station.

The student has typed the desired procedure at the Student Station.

The teacher turns the knob of the Network 2 to the number that corresponds with that Student Station.

At the Host computer, the teacher types,

(BREAK) (L) (T) (ENTER)

(to prepare the Host computer to receive the procedure through the Network 2)

At the Student Station, the student types,
(BREAK) (S) (T) (ENTER)
(to send the procedure through the Network 2 to the Host computer)

The student's procedure is now transferred to the teacher's Host computer. In this example, the teacher may want to save it as procedure "G" on the Host diskette.

At the Host computer, the teacher types,
(S) (G) (ENTER)

The student's LOGO procedure is now saved as procedure "G" on the teacher's diskette.

Note that only one disk-equipped Color Computer is used in this classroom, but the entire class can save their LOGO procedures on diskettes. This provides several advantages. Procedures from sixteen different students can be saved quickly and reliably on diskette, and can be located and re-loaded easily. Saving procedures on a central disk also keeps control in the hands of the teacher, and saves a lot of time that would otherwise be spent in supervising the use of up to sixteen computer cassette recorders.

THE COST OF THE NETWORK 2 COLOR LOGO LAB

Hardware Costs		
1	32K TRS-80 Color Computer with Extended Color BASIC (the Host computer)	$ 549.95
1	Color TV (for the Host computer)	$ 399.95
1	Disk Drive for the Host computer	$ 599.00
1	Network 2 Controller	$ 499.00
16	16K TRS-80 Color Computers with standard Color BASIC (16 x 299.95)	$ 4,799.20
16	Color TVs (for the Student Stations) (16 x 399.95)	$ 6,399.20
Software Costs		
1	Color LOGO (Disk Version)	$ 99.00
16	Color LOGO (Program Paks) (16 x 49.95)	$ 799.20
	Total Cost	$14,144.50

This represents the complete cost of the Network 2 Color LOGO Lab, including the TV monitors.

Public schools and some private schools are eligible for discounts on this, and other Radio Shack hardware and software. For discount information and other information, call or write Radio Shack's Regional Educational Coordinator for your area.

Radio Shack Regional Educational Coordinators

Albany Region:
Don Francolino
Radio Shack
39 S. Main Street
W. Hartford, CT 06107
203/232-4529

Atlanta Region:
Art Williams
Radio Shack
241 W. Wieuca Rd., N.E.
Atlanta, GA 30342
404/255-9438

Boston Region:
Dick Callahan
Radio Shack
Education Division
250 Granite Street
Braintree, MA 02368
617/848-0780

Chicago Region:
Donna Comber
Radio Shack
679 W. North Ave., Suite 204
Elmhurst, IL 60126

Columbus Region:
Sonny O. Compton
Radio Shack
4343 Williams Road
Groveport, OH 43125
614/836-2373

Dallas Region
Sid Agent
Radio Shack
2588 Royal Lane
Dallas, TX 75229
214/484-9943 (4) (5)

Denver Region:
Rosemary Shiels
Radio Shack
5890 W. 44th Avenue
Denver, CO 80212
303/424-4467

Detroit Region:
Celia Magro
Radio Shack
29548 Southfield Rd., Suite 200
Southfield, MI 48076
313/552-9290

Houston Region
Jim Savoie
Radio Shack
7119 San Pedro
San Antonio, TX 78216
512/341-2622

Kansas City Region:
Dan Hennessey
Radio Shack
9400 36th Avenue, North
Minneapolis, MN 55427
612/546-4888

Los Angeles Region:
Terry Kramer
Radio Shack
14126 E. Firestone
Santa Fe Springs, CA 90670
213/921-2659

Louisville Region:
Penny Shattuck
Radio Shack
553 N. Court, Suite 175
Palatine, IL 60067
312/991-2275

Miami Region:
Peter Lenkway
Radio Shack
4360 N. Federal Highway
Ft. Lauderdale, FL 33308
305/493-5021

New York Region:
Bob Sochor
Radio Shack
6 East 39th Street, Room 401
New York, NY 10016
212/696-9800

Mick Nuspl
Radio Shack
6 East 39th Street, Room 401
New York, NY 10016
212/696-9800

Norfolk Region:
Jerry Proffitt
Radio Shack
5701 West Park Drive,
Suite 205
Charlotte, NC 28210
704/527-7442

Philadelphia Region:
Donald Wallick
Radio Shack
Rt. 130 & Cinnaminson Ave.
Cinnaminson, NJ 08077
609/829-6911

Madelyn Lesnick
Radio Shack
Crest Plaza
Cedar Crest Blvd. & U.S. 22
Allentown, PA 18104
215/395-5755

Pittsburgh Region:
Winston Ferrell
Radio Shack
1679 Washington Road
Pittsburgh, PA 15228
412/833-1918

San Diego Region:
Bob Norman
Radio Shack
12821 Knott Street
Garden Grove, CA 92641
714/894-1371 x255

San Francisco Region:
Steve Terhune
Radio Shack
1291 E. Hillsdale, Suite 211
Foster City, CA 94404
415/574-1708

Seattle Region:
Annie Gillvan
Radio Shack
5030 Roosevelt Way, N.E.
Seattle, WA 98105
206/527-0940

Tampa Region:
Paul Hoagland
Radio Shack
5729 N. Hoover Blvd., Suite 230
Tampa, FL 33614-5373
813/886-2974

Washington, D.C. Region
Kevin Hogans
Radio Shack
Computer Center #7267
48 W. Ridgley Road
Lutherville, MD 21093
301/561-2474

Eastern Educational Marketing Manager:

Ron Moore
Radio Shack
1400 One Tandy Center
Fort Worth, TX 76102
817/390-3910

Western Educational Marketing Manager:

Scott Bowers
Radio Shack
1400 One Tandy Center
Fort Worth, TX 76102
817/390-3910

TRS-80 Model 4 Desktop Computer

By Bruce Elliott

A special note to owners of Models I and III:

The new Model 4 may look similar to the Model III with an attractive off-white finish, but the differences are amazing. Here are some of the things which you have asked for that are in the Model 4: 64K RAM machine (CP/M®) compatible, 80 column by 24 line display, Audio, Three special function keys, a Control Key, Caps key, and a new TRSDOS which is data file compatible with LDOS 5.1.3. Read on for more information on this incredible machine. . .

The Model 4 is a Desktop Computer designed for professionals, managers, educators and personal computer users. The two disk version is priced at $1999 and includes a Z80A operating at 4 MHZ, 64K RAM, two double density disk drives, parallel printer interface, RS232C interface, 12" Monitor, 70 key keyboard with numeric pad and an 80 column by 24 line display providing upper and lower case characters and reverse video. Sound (1 voice, 7 tone) is also included.

Many of the features mentioned above are "extras" with other popular computers. A comparably configured Apple IIe "package" for example, would still require the addition of a numeric pad, 80 column/memory upgrade, second disk drive, hardware card containing Z80 for CP/M compatibility, RS232C and printer interface, a communications program, print spooler option, memory disk option and additional documentation. An IBM® PC would typically require the additional purchase of MSDOS™, a second drive, monitor, monochrome adapter card, asynchronous adapter card, and a Z80 option card for CP/M compatibility. (Some options provided by third party suppliers.)

The Model 4 two disk version provides a complete system (less printer) and includes in the base price the popular options mentioned above which personal computer users typically add to their system after purchase. Additional options for the Model 4 include an additional 64K RAM, hard disk, a large selection of printers and other peripherals including, pen/plotters, digitizers, and more. CP/M Plus and bit mapped high resolution graphics will be available soon.

The Model 4 includes a new TRSDOS and an all new Microsoft Disk BASIC, along with a tutorial introduction to programming in BASIC, an introduction to the computer, a 350 page Reference Manual, and a reference card. The Model 4 provides access to a large library of software from Radio Shack and other suppliers since it supports Model III TRSDOS, LDOS, C/PM PLUS, and its own TRSDOS version 6.0.

Feature/Benefits Summary:

DISPLAY

An 80 column display is **standard** on any disk based Model 4. (The tape based 16K Model 4 works just like our Model III, including 64 × 16 screen.) The 80 character by 24 line display will be supported by new Model 4 disk software for word processing, spreadsheets, and more. In addition, a double width or 40 character display mode is software selectable. Model 4 also operates in 64 characters by 16 lines or double width 32 by 16 display selections under TRSDOS 1.3 or LDOS 5.13. These widths provide compatibility with the vast library of Model III and Model I software.

Reverse video is included along with the ability to generate block graphics, and 96 special graphics characters. An optional bit mapped high resolution graphics board and software package will be available to provide 640 by 240 pixel resolution, along with a Graphics BASIC that provides easy to use commands like PAINT, LINE, CIRCLE, and others.

Model 4 includes a 12" high-resolution B/W monitor enclosed in a compact (12 1/2" by 18 7/8" by 21 1/2") Desktop Unit providing for internal expansion.

User selectable options include choice of cursor character from the ASCII character set or special character set, and blinking/non-blinking cursor.

MEMORY

Expansion to 128K bytes of main memory. 64K RAM is standard for Model 4 disk systems with an additional 64K RAM option (for a total of 128K) priced at $149.00 (26-1122 plus installation). Radio Shack will introduce new products, including VisiCalc, that will take advantage of the additional memory.

Model 4 TRSDOS includes a "MEMDISK" feature that easily simulates a disk drive in available memory, either within the standard 64K, or using the additional 64K "bank" of memory. This allows owners to treat their memory disk drive (or MEMDISK) like any other disk drive, but gain significant speed advantages. No hardware options are required, and the MEMDISK software is included with the Model 4 disk versions. MEMDISK is great for programs which use overlays, or program modules normally residing on disk. These files (or TRSDOS system files) can be placed in MEMDISK for amazing response.

Cassette systems are upgraded to 64K from 16K with the same memory option (26-1122) at $149.00 plus installation. Disk drives are available for upgrading the cassette version. The first internal drive is now $649.00 plus installation (26-1127). This price includes TRSDOS 6.0, Disk BASIC, the sound feature, and the disk system documentation.

SPEED

The Model 4 uses a Z80 A operating at a 4 MHZ clock rate for high performance. The Model III computer (and the Model 4 operating as a Model III) operate at 2 MHZ. TRSDOS 6.0 is designed with high performance in mind. The combination of TRSDOS 6.0 and the Z80 A provide for fast operation. When combined with MEMDISK, owners will find that the Model 4 can tackle big jobs easily.

VAST SOFTWARE LIBRARY

The Model 4 executes, without change, all Radio Shack software available for the TRS-80 Model III. No conversion or

updating is required since the Model III operating system (TRSDOS 1.3) will run unaltered in the Model 4. Operation of Model III software is possible because we included the Model III ROM in the Model 4. The ROM is only used for Model III software, and we have tested the Model III software to ensure compatibility. As a result, the Model 4 can operated in two "modes" looking like either a TRS-80 Model III, to gain access to current software, or running at the default 4 MHZ speed and 80 column display when utilizing Model 4 TRSDOS 6.0.

Other options for the Model 4 include CP/M. Radio Shack will sell CP/M Plus and CBASIC. C/PM Plus is the latest version of CP/M and is compatible with CP/M 2.2, but has many new features. Both are provided to Radio Shack by Digital Research. No hardware card is required and the user simply boots the CP/M version of the software.

Radio Shack's current alternate operating system for the Model III, LDOS 5.1.3 will also operate unaltered in the Model 4.

As a result of these "modes", thousands of programs are currently available for the TRS-80 Model 4. This allows current owners of the TRS-80 Model III to obtain both data file compatability and media interchangeability by using LDOS on their Model IIIs, and TRSDOS version 6 on their Model 4s. Model I owners, operating under LDOS, are able to interchange data files with Model IIIs using LDOS, and Model 4s using TRSDOS version 6.

Model 4 owners simply use current Model III application disks as they are, or run new Model 4 applications designed to take advantage of the increased speed and new features of the Model 4.

KEYBOARD

The Model 4 has a 70-key typewriter-style keyboard with a standard 12-key numeric pad. Now included are 3 function keys, caps and control keys for use with TRSDOS 6.0. The keyboard can generate the full ASCII character set. The keyboard delay (the length of time before a key begins repeating) and repeat rates are user definable from the operating system. TRSDOS 6.0 also provides a keyboard type-ahead feature and the use of a "keyclick" filter (so you can hear each key as its is pressed) when desired by the operator. In addition, TRSDOS 6.0 allows the user to assign strings to the alphabetic keys. This feature allows often entered TRSDOS commands or BASIC commands to be stored and then accessed with a simple keystroke to speed programming.

SOUND

Sound is provided with Model 4 Disk systems and Disk upgrade kits. The sound's tone and duration are easily specified, without POKEs, using BASIC's SOUND command. In addition, TRSDOS provides for use of the sound feature in JCL files to prompt an operator during unattended execution.

THE SYSTEMS

The Model 4 cassette system (RS stock #26-1067, $999.00) includes 16K RAM, with internal expansion room to add 2 disk drives, more RAM (up to 128K total), RS232C, and the high-resolution option.

The single drive Model 4 (RS stock #26-1068, $1699.00) includes 64K RAM and one 40 track double density 5¼" disk drive providing 174K formatted capacity.

The Model 4 two drive Business Computer (RS stock #26-1069, $1999.00) provides 64K RAM and two 40 track double density 5¼" drives with a total disk capacity of 348K. The two drive system also includes RS232C.

All disk based Model 4s allow for the addition of two more external drives and up to 4 of Radio Shack's 5 MB Hard Disk units.

Disk Systems include Model 4 TRSDOS version 6.0 and DISK BASIC. Both include advanced features designed for personal or business use.

MODEL 4 TRSDOS FEATURES INCLUDE:

All files carry the date they were created and carry a flag if they have been modified. This makes selecting files to be backed up, removed, or copied quite easy. TRSDOS 6.0 even allows you to partially describe the names in a group of the files rather than using each filename.

These file selection features speed backing up hard disk files to floppy disk. When using a hard disk, backups can span multiple floppies, and TRSDOS will prompt you for diskette swaps when reconstructing a file on the hard disk.

Directory listings are sorted alphabetically by filename, and can be further specified by date range, whether modified or not, and by partial description of the filename.

TRSDOS contains a complete Job Control language to handle unattended operations and completely control the application. Execution can be tied to the time using the WAIT and SLEEP commands, and can provide both visual and audio alerts. JOBLOG records the results of an unattended application including a time stamp for audit purposes or error history.

Cassette support is provided for the Model 4 to read or write cassette tapes made with the new TRS-80 Model 100.

TRSDOS 6.0 includes COMM, a complete terminal program which includes disk file send and receive, and Radio Shack modem support including 1200 baud for communication with other TRS-80s or information networks. Automatic buffering and control of handshaking is available in COMM, if required.

The SPOOL command allows owners to keep using the Model 4 even while printing, by assigning available memory or disk space as a "buffer" to capture output. This keeps the printer busy while you do something else. Your SPOOL memory can come from the regular 64K RAM, or you may assign 32K of the additional bank of memory to the spooler.

The FORMS command lets you tailor printouts and handle preprinted forms by changing the number of lines per page, set margins, set number characters per printed line, set page length, indent, and more.

TRSDOS provides the ability to LINK devices to one another, or to ROUTE information to new destinations. These features add flexibility to your system. For example, you might send program output to disk rather than to a printer when the printer is not available, or you might transmit normal printed output, like a report, through the RS232C in addition to printing it.

Input/output data can be FILTERed or modified by a user program to aid an application. Examples would be to alter printer output, RS232C output, act upon certain keys when pressed, as determined by a user driver or FILTER program.

Included are a disk altering DEBUG utility with simultaneous Hex and ASCII display, TIME and DATE display, PATCH, and other disk utilities.

Model 4 BASIC is a new Microsoft BASIC. The BASIC includes CHAIN and COMMON to allow larger BASIC programs and 40 character maximum variable names. Model III owners will still be able to run their BASIC applications in the Model III mode (under TRSDOS 1.3) without changes.

Model III UPGRADE TO MODEL 4

Radio Shack will upgrade your Model III to include comparable Model 4 capabilities for only $799 (26-1123), plus installation.

If you have a TRS-80 Model III with at least one disk drive and 32K RAM, your Model III will be upgraded to a 64K Model 4. We will replace your Model III keyboard with a Model 4 keyboard, and we will install Model 4 sound capabilities. You will retain the same Model III cabinet, disk drive(s), and RS-232 (if it was in your Model III). We will also include a new Model 4 logo, 64K memory button, TRSDOS 6.0, Disk BASIC and new documentation.

This upgrade is done by a board exchange. This means that Radio Shack will put a Model 4 board and memory into your machine and keep the old Model III board and memory.

Your newly upgraded Model III will still be able to do all the things it could do as a Model III, as well as have the ability to run TRSDOS 6.0 and optionally CP/M Plus.

MODEL 4 BASIC

Unless otherwise stated, the following information applies to Model 4 BASIC.

40 character variable names (That is, the first forty characters in a variable name are considered significant. In Model III BASIC, only the first two characters in a variable's name are significant.)

Below is a list of BASIC statements and commands from Model II/12 BASIC, Model III Disk BASIC, and Model 4 Disk BASIC. Model II/12 BASIC was included in the list for comparison purposes.

Statements for assigning values to variables and defining memory space

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CLEAR	X	X	X	Clears the value of all variables and closes all open files. Can optionally specify the highest memory location available to BASIC and specify the amount of memory available for stack space. In the Model 4, string space is managed dynamically and is not specified in the CLEAR statement. Model III BASIC: CLEAR is used to reset all numeric variables to zero, and all string variables to null. Optionally, CLEAR can specify the number of bytes available for string storage.
COMMON			X	Passes one or more variables to a chained program.
DATA	X	X	X	Stores numeric and string constants to be accessed by a READ statement.
DEF FN	X	X	X	Defines a function name based on the following function definition.
DEFDBL	X	X	X	Defines any variable beginning with the specified letters as being double precision.
DEFINT	X	X	X	Defines any variable beginning with the specified letters as being integer.
DEFSNG	X	X	X	Defines any variable beginning with the specified letters as being single precision.
DEFSTR	X	X	X	Defines any variable beginning with the specified letters as being a string.
DIM	X	X	X	Sets aside storage for the specified arrays with the dimensions given. All array values are set to zero for numeric arrays, or to null for string arrays.
ERASE	X		X	Erases one or more arrays from a program.
LET	X	X	X	Assigns the expression value to the specified variable. The keyword LET is optional.
MID$	X	X	X	Used on the left side of an assignment statement, MID$ replaces a portion of the existing string with a new string.
OPTION BASE			X	The value supplied (0 or 1) determines the minimum value for an array subscript. The default minimum is zero.
RANDOM	X	X	X	Reseeds the pseudo-random number generator.
READ	X	X	X	Reads values from a DATA statement and assigns them to supplied variables.
RESTORE	X	X	X	Restores a program's access to previously read DATA statements. Line number is an optional parameter.
SWAP	X		X	Exchanges the values of two variables of the same type.

Statements for altering program sequence

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CHAIN			X	Loads a BASIC program, chains it to a "main" program, and begins running it. Options include MERGE, first line number to be executed, variable value retention without COMMON, and DELETE specified range of program lines.
END	X	X	X	Ends executions of a BASIC program.
FOR/STEP/ NEXT	X	X	X	Establishes a program loop. Note: Model 4 BASIC skips the body of the loop if the loop's initial value times the sign of the STEP increment exceeds the loop's final value times the sign of the STEP increment. This prevents the computer from inadvertantly entering an endless loop.
GOSUB	X	X	X	Goes to a subroutine which begins at the line specified.
GOTO	X	X	X	Goes to the specified line.
IF/THEN/ ELSE	X	X	X	Tests a conditional expression and modifies program flow based on the result of the test.
ON . . . GOSUB	X	X	X	Goes to a subroutine at a line number which is determined by the value of a supplied number.
ON . . . GOTO	X	X	X	Goes to a line number which is determined by the value of a supplied number.
RETURN	X	X	X	Returns program control to the BASIC statement immediately following the most recently executed GOSUB.
STOP	X	X	X	Stops program execution.
WHILE . . . WEND			X	Executes a series of program statements in a loop as long as a given condition is true.
WAIT			X	Suspends program execution until a machine input port contains a specified value.

Statements for storing and accessing data on a disk

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CLOSE	X	X	X	Closes access to a disk file.
FIELD	X	X	X	Divides a direct-access buffer into one or more fields.
GET	X	X	X	Gets a record from a direct access disk file and places it in the specified buffer.
INPUT#	X	X	X	Inputs data from a sequential disk file and stores it in numeric and/or string variables. Leading spaces in the data file are ignored.
LINE INPUT#	X	X	X	Inputs an entire line from a sequential disk file into a string variable.
LSET	X	X	X	Sets data, left justified, into the specified field in a direct-access buffer.
OPEN	X	X	X	Opens a disk file. The information supplied will determine if the file will be used for sequential input, sequential output, or direct access. Also specified is the file buffer to be used, the name of the file, and optionally, the record length for direct access.

PRINT#	X	X	X	Prints data items into a sequential file using the specified disk file buffer.
PRINT# USING	X	X	X	Prints data items into a sequential file using the specified disk file buffer in the format specified by the USING statement.
PUT	X	X	X	Puts a record into a direct-access file using the indicated disk file buffer.
RSET	X	X	X	Sets data, right justified, into the specified field in a direct-access buffer.
WRITE#			X	Writes data to a sequential access disk file.

Statements for debugging a program

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CONT	X	X	X	Resumes program execution after (BREAK) or STOP.
ERL	X	X	X	Returns the line number in which the most recent error occurred.
ERR	X	X	X	Returns the error code of the most recent error. Model III BASIC: Use ERR/2 + 1 to determine the actual error code.
ERR$			X	Returns a system error number and message.
ERROR	X	X	X	Simulates the specified error during program execution.
ON ERROR GOTO	X	X	X	Transfers program control to the indicated line if an error occurs.
REM	X	X	X	Inserts a remark line in a program.
RESUME	X	X	X	Tells the program where to continue execution after an error has been handled by the program.
TROFF	X	X	X	Turns the trace function off.
TRON	X	X	X	Turns the trace function on.

Statements for inputting or outputting data to the video display or the line printer

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CLS	X	X	X	Clears the screen.
INPUT	X	X	X	Inputs string and/or numeric data from the keyboard into one or more variables. An optional string prompt may also be displayed.
LINE INPUT	X	X	X	Inputs an entire line (up to 254 characters) from the keyboard.
LIST	X	X	X	Lists the BASIC program in memory to the display.
LLIST	X	X	X	Lists BASIC program lines in memory to the printer.
LPRINT	X	X	X	Prints data to the printer if one is available. LPRINT can be used in conjunction with TAB and USING.
PRINT	X	X	X	Prints numeric or string data on the display.
PRINT@	X	X	X	Prints numeric or string data on the display at the specified screen position. The screen position can be specified as an absolute value (0 - 1919) or by indicating the screen row (0 - 23) and column (0 - 79) at which the data item is to be placed. Model III BASIC does not support PRINT@ with row and column. Also, there are only 1024 character positions on (0 - 1023) on the Model III video.
PRINT TAB	X	X	X	Prints numeric or string data on the display at the horizontal position specified by the TAB value.
PRINT USING	X	X	X	Prints numeric or string data on the display according to the format specified in the USING statement.
WIDTH			X	Sets the line width for printing on the display or on a printer.
WRITE			X	Writes data on the display.

Statements for performing system functions or entering other modes of operation

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
AUTO	X	X	X	Automatically generates new line numbers during program entry. Optionally, the starting line number and the increment between successive line numbers may be specified.
CALL			X	Transfers program control to an assembly-language subroutine stored at variable. Variable contains the starting address for the subroutine. An optional parameter list can be included to pass values to the subroutine.
CMD "A"			X	Return to Model III TRSDOS.
CMD "B"			X	Enable/Disable the (BREAK) key.
CMD "C"			X	Compress a BASIC program by removing spaces and REMarks.
CMD "D"			X	Display the directory for the specified drive.
CMD "E"			X	Display the previous TRSDOS error.
CMD "I"			X	Return a command to TRSDOS for execution.
CMD "J"			X	Convert a calendar date.
CMD "L"			X	Load a Z-80 subroutine or program file into RAM.
CMD "O"			X	Sort a string array.
CMD "P"			X	Check printer status.
CMD "R"			X	Start real-time clock display.
CMD "S"			X	Normal return to TRSDOS.
CMD "T"			X	Turn off real-time clock display.
CMD "X"			X	Cross-reference of reserved words, string variables, or strings in a BASIC program.
CMD "Z"			X	Duplicate output to display and the printer.
DELETE	X	X	X	Deletes specified lines from the BASIC program in memory.
DEF USR	X	X	X	Defines the starting address for one of 10 (0-9) assembly language routines.
EDIT	X	X	X	Enters the EDIT Mode so you can edit the specified line.
KILL	X	X	X	Deletes the specified file from disk.
LOAD	X	X	X	Loads the specified BASIC program into memory from disk. Any previously resident BASIC program will be deleted, all variables will be cleared, and all OPEN files will be closed. The ",R" option will leave OPEN files open and automatically RUN the program.
MERGE	X	X	X	Loads the specified BASIC program into memory and merges it with the BASIC program currently in memory.
NAME	X	X	X	Allows you to rename a disk file without returning to TRSDOS. Model III Disk BASIC: The keyword NAME is used for the renumber function.
NEW	X	X	X	Deletes the BASIC program currently in memory and clears all variables.
OUT		X	X	Sends the supplied data byte to the specified machine output port.
POKE		X	X	Writes the supplied data byte into a memory location.
RENUM	X	X	X	Renumbers a BASIC program. Optional parameters include line number to start renumbering at, first new line number to use, and the increment between new line numbers. In Model III Disk BASIC the keyword NAME is used for the renumber function.
RUN	X	X	X	Runs a program. RUN by itself, or with an optional line number, will execute the BASIC program currently in memory. RUN followed by a disk filespec and optionally by a ',R' will load the indicated BASIC program from disk into memory and execute it. The optional ',R' will cause disk files, which were OPEN before the RUN statement was executed, to remain open.
SAVE	X	X	X	Saves the BASIC program currently in memory onto disk in compressed format with the provided filename. The ',A' option will save the

				program in ASCII format. The ',P' option protects the file by saving it in an encoded binary format. Model III BASIC does not have the ',P' option.
SYSTEM	X	X	X	Returns the computer to TRSDOS. If an optional library command is included, the command will be executed by TRSDOS, and you will be returned to BASIC with your program and variables intact. Model III BASIC: The SYSTEM command is used to put the computer into the System mode, which allows you to load object files from tape.

Numeric Functions (Unless otherwise specified, numeric functions return single precision results.)

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
ABS	X	X	X	Computes the absolute value of a number.
ASC	X	X	X	Returns the ASCII code for the first character of a string.
ATN	X	X	X	Computes the arc tangent (in radians) of a number.
CDBL	X	X	X	Converts a number to double precision.
CINT	X	X	X	Converts a number to integer representation.
COS	X	X	X	Computes the cosine of a number whose value is in radians.
CSNG	X	X	X	Converts a number to single precision.
EXP	X	X	X	Calculates the natural exponential of the specified number.
FIX	X	X	X	Returns the truncated integer form of the specified number.
FRE	X	X	X	Returns the amount of free memory space (if a dummy number is used) or the amount of free string space (if a dummy string is used.)
INSTR	X	X	X	Searches for the first occurrence of the search string in the target string. An optional character position for starting the string may be specified.
INP		X	X	Inputs a value (0-255) representing the information present at the specified port.
INT	X	X	X	Converts the specified number to the largest integer which is not larger than the specified number. The value of the specified number is not limited to the normal integer range.
LEN	X	X	X	Returns the number of characters in the string.
LOG	X	X	X	Computes the natural logarithm of the number.
MEM	X	X	X	Returns the number of unused and unprotected bytes in memory.
PEEK		X	X	Returns the value of the specified memory location.
RND	X	X	X	Generates a pseudorandom number between zero and the supplied number. If the supplied number is zero, a decimal value between 0 and 1 will be generated.
SGN	X	X	X	Determines the algebraic sign of the number.
SIN	X	X	X	Computes the sine of a value provided in radians.
SQR	X	X	X	Calculates the square root of a number.
TAN	X	X	X	Computes the tangent of a number supplied in radians.
USR	X	X	X	Calls a user's assembly-language subroutine and passes a value to the routine.
VAL	X	X	X	Calculates the numeric value of a string.
VARPTR	X	X	X	Returns the absolute memory location where a variable or disk buffer is stored in memory.

String functions

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CHR$	X	X	X	Returns the character corresponding to an ASCII or control code.
DATE$	X		X	Returns the current system date.
ERRS$	X		X	Returns the number and description of the most recent BASIC disk-related error.
HEX$	X		X	Calculates the integer hexadecimal value of the number specified, and returns this value as a string.
LEFT$	X	X	X	Returns all characters left of and including the specified position in the string.
MID$	X	X	X	Returns a substring of the specified string, beginning at the indicated character position.
OCT$	X		X	Calculates the integer octal value of the number specified, and returns this value as a string.
RIGHT$	X	X	X	Returns the rightmost number of characters in a string.
SPACE$	X		X	Returns a string of spaces. The length of the string is determined by the value provided.
STR$	X	X	X	Converts the number to a string value.
STRING$	X	X	X	Returns a string containing a specified number of identical characters.
TIME$	X	X	X	Returns the time of day as set in the computer. Model III BASIC: Returns both the system time and date.

Input/Output Functions

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CVD	X	X	X	Convert string value to a double precision numeric value.
CVI	X	X	X	Convert string value to an integer.
CVS	X	X	X	Convert string value to a single precision numeric value.
EOF	X	X	X	Detects the end of the specified disk file.
INKEY$	X	X	X	Returns a one-character string from the keyboard without the user having to press (ENTER). If no key is pressed, a null string is returned.
INPUT$	X		X	Inputs a string of characters from the keyboard. If the optional buffer number is present, the string of characters is input from the specified sequential disk file.
LOC	X	X	X	Returns the current record number for the specified disk file.
LOF	X	X	X	Returns the number of the last record in a direct access file.
LPOS			X	Returns the position of the line printer's print head within the line printer's buffer. Note: The value returned by LPOS does not necessarily give the physical position of the print head.
MKD$	X	X	X	Converts a double precision numeric value to a string value. Used with LSET and RSET to place a double precision number into a direct access file buffer.
MKI$	X	X	X	Converts an integer value to a string value. Used with LSET and RSET to place an integer into a direct access file buffer.
MKS$	X	X	X	Converts a single precision numeric value to a string value. Used with LSET and RSET to place a single precision number into a direct access file buffer.
POS	X	X	X	Returns the current horizontal (column) position of the cursor on the display.
ROW	X		X	Returns the current vertical (row) position of the cursor on the display.
SPC	X		X	Prints a line of spaces. The number of spaces printed is determined by the value provided. SPC can only be used with PRINT, LPRINT, and PRINT#.

Statements for storing and accessing data on a tape.

Statement	Model II BASIC	Model III Disk BASIC	Model 4 Disk BASIC	Brief Description
CLOAD		X		Lets you load a BASIC program stored on cassette.

CLOAD?	X	Lets you compare a BASIC program stored on cassette with the program currently stored in memory.
CSAVE	X	Stores the memory resident BASIC program on cassette tape.
INPUT#-1,	X	Inputs the specified number of variables stored on cassette tape and assigns them to the specified variable names.
PRINT#-1,	X	Prints the values of the specified variables onto cassette tape.

Statements for Model III graphics.

Statement	Model II BASIC / Model III Disk BASIC / Model 4 Disk BASIC	Brief Description
POINT	X	Tests the specified graphics block to determine if it is 'on' or 'off'.
RESET	X	Turns off a graphics block at the location specified by the given coordinates.
SET	X	Turns on the graphics block at the location specified by the given coordinates.

DISK OPERATING SYSTEMS

The Model 4 will operate under three different operating systems. If you use a Model III TRSDOS 1.3 diskette (not included) in the machine, the Model 4 will behave as a Model III, including the 64 character Model III screen and the 2Mhz clock rate. If you use your Model 4 TRSDOS 6.0 diskette, the Model 4 will use its 80 column video, and the other features of Model 4 TRSDOS. Coming soon will be Radio Shack's CP/M Plus for the Model 4.

Below is a chart which describes the operating system commands available under the various Model I (TRSDOS 2.3, 2.7DD and LDOS 5.1.3), Model III (TRSDOS 1.3 and LDOS 5.1.3), and Model 4 (TRSDOS 1.3 and 6.0) operating systems. LDOS 5.1.3 is used with Model I and III. Data files on LDOS 5.1.3 can be read directly by TRSDOS 6.0, and TRSDOS 6.0 data files are readable by LDOS 5.1.3 (if your Model I is not equipped with Radio Shack double density, you may need to use single density on the Model 4.)

Command	TRSDOS 2.3	TRSDOS 2.7DD	TRSDOS 1.3	LDOS 5.1.3	TRSDOS 6.0	Explanation
APPEND	X	X	X	X	X	Copies the contents of the source file onto the end of the destination file. TRSDOS 6.0 includes two optional parameters, ECHO, which echoes the characters to the screen while appending, and STRIP, which removes the last byte from the destination file before appending. This STRIP feature is useful for appending files which contain end of file markers like SCRIPSIT.
ATTRIB	X	X	X	X	X	Assigns protection attributes to a particular file or group of files on a particular drive.
AUTO	X	X	X	X	X	Stores an AUTO command line on the disk. This command line automatically executes whenever you start up or reset TRSDOS.
BACKUP	X	X	X	X	X	Duplicates (backs up) all or some of the files from the source disk to the destination disk. TRSDOS 6.0 and LDOS 5.1.3 use a parameter list to tell the system which file or group of files to duplicate.
BASIC	X	X	X		X	Disk BASIC.
BASIC2	X					Jump to Model I Level II BASIC.
BASICR	X					Load and RUN Enhanced Disk BASIC.
BLINK		X				Turn Model I blinking cursor on or off.
BOOT				X	X	Resets (boots) the system by returning the system to its original start-up condition. The use of optional keys can prevent execution of AUTO or SYSGEN configurations, as well as invoking the system debugger.
BUILD		X	X	X	X	Lets you enter data (such as commands) and save these to disk. BUILD is used primarily to create files for DO, PATCH and, in TRSDOS 6.0, KSM and FLT. Optional parameters in TRSDOS 6.0 permit BUILD files to be extended, or to contain data in HEX format.
CLEAR		X	X	X	X	Clear user memory. See MEMORY for LDOS 5.1.3 and TRSDOS 6.0.
CLOCK	X	X	X	X		This command turns the clock screen display on or off.
CLS		X	X			Clear the screen.
CLICK /FLT					X	A keyboard filter to give the keys a "click" sound when they are pressed.
CMDFILE				X		A general purpose disk-to-disk, tape-to-disk, disk-to-tape utility for moving machine language files from one media to another.
COMM					X	Lets two computers communicate via a defined device. COMM is a full featured communications program similar to LCOMM in LDOS 5.1.3 and TERMINAL on the Model 12.
CONFIG		X				Change disk's number of tracks.
CONV				X	X	Allows you to move (convert) data files from a Model III TRSDOS 1.3 diskette to a TRSDOS 6.0 formatted diskette, and to view 1.3 directories from L.O.
CONVERT			X			Convert Model I disk files to Model III diskettes.
COPY	X	X	X	X	X	Copies information from one disk or device to another disk or device. TRSDOS 6.0 gives you the ability to change the logical record length of the information during the COPY process.
COPY23B /BAS				X		Copies Model I TRSDOS 2.3B files to LDOS 5.1.3 diskettes.
CREATE		X	X	X	X	Creates a file on disk and pre-allocates disk space for its contents.
DATE	X	X	X	X	X	Resets the date or displays the current system date.
DEBUG	X	X	X	X	X	Turns the debug utility on or off.
DEVICE	X			X	X	Displays the status of the drives, selected options, and data paths for logical devices which have been set, routed, or linked. Model I TRSDOS 2.3—simply lists all currently defined I/O devices.
DIR	X	X	X	X	X	Displays the specified disk's directory.
DISKDUMP /BAS	X					Examine a disk file.
DO		X	X	X	X	Compiles and executes a DO file. A DO file, in LDOS 5.1.3 and TRSDOS 6.0, is a user created Job Control Language (JCL) file that contains one or more library commands or file specifications.
DUAL		X	X	*	*	Duplicates output to both video and a printer. *Use link in LDOS 5.1.3 and TRSDOS 6.0.
DUMP	X	X	X	X	X	Copies an area of memory to a named disk file.
ERASE		X				Erase a file from disk by zeroing the disk data area.
ERROR		X	X			Display a full error message.
FILFIX		X				Load and modify the content of a file.
FILTER				X	X	Filters or modifies data as it is read from or written to a device.
FORMAT	X	X	X	X	X	Prepares a blank or old disk for use by defining the tracks and sectors, and by writing system information onto the disk.
FORMS /FLT			X	X	X	Sets up printer filter parameters. (PR/FLT in LDOS 5.1.3 and FORMS in TRSDOS 1.3.)

FREE	X	X	X	X	X	Lists the amount of free space and the number of files on each drive. If a drive number is specified, FREE displays a space map of the disk in that drive.
GETDISK /BAS	X					Copies a disk file to tape.
GETTAPE /BAS	X					Copies a tape data file to disk.
HELP		X	X			Provides a brief explanation of a command's syntax.
HERZ50			X			Set up the Model III for 50hz AC power.
HITAPE				X		The HITAPE utility is for Model III operations only. It permits the use of high speed cassette operations with LBASIC and CMDFILE.
JCL				X	X	Job Control Language is one of the most powerful features of TRSDOS 6.0 and LDOS 5.1.3. It allows the user to construct a sequence of commands and statements to control the actions of the operating system or applications programs.
JL/DVR				X		JOBLOG creates a log of all commands entered or received, along with a time stamp for each. JOBLOG is established using the JL/DVR driver.
KI/DVR				X		KI/DVR enables keyboard features like type ahead and JKL screen print.
KILL	X	X	X	X		See REMOVE
KSM/FLT				X		KSM/FLT allows use of files containing phrases associated with the unshifted alphabetic keys to be used as direct keyboard inputs.
LBASIC				X		The disk version of BASIC under LDOS 5.1.3
LCOMM				X		See COMM
LIB	X	X	X	X	X	Displays a listing of all available system commands. TRSDOS 6.0 has three separate libraries of commands.
LINK				X	X	Links together two logical devices. Both devices must be enabled in the system.
LIST	X	X	X	X	X	Lists the contents of a file. Optional parameters determine exactly what information will be displayed.
LOAD	X	X	X	X	X	Loads a machine language program file into memory without executing that file.
LOG				X	X	Logs in the directory track and number of sides on a diskette. In TRSDOS 6.0 LOG is used for hard disk installations.
LPC		X	X			Line printer control program.
MASTER		X	X			Set the master read/write drive. SYSTEM performs this function in LDOS 5.1.3 and TRSDOS 6.0.
MEMDISK /DCT					X	Allows you to set aside some portion of your available RAM memory to function as a disk drive. MEMDISK can be used with the extra 64K RAM in a 128K Model 4, or memory can be set aside in the normal 64K area.
MEMORY				X	X	Allows you to reserve a portion of memory, display the current HIGH$ and LOW$, modify a memory address, or jump to a specified memory location.
MEMTEST		X	X			Memory test routine for ROM and RAM.
MINIDOS /FLT				X		The MiniDOS filter program provides a means to access certain LDOS functions without having to be at the LDOS Ready prompt.
MODx/DCT				X	X	The MOD1 and MOD3/DCT programs are used to change the logical drive numbers of 5" floppy drives. FLOPPY/DCT in TRSDOS 6.0.
PATCH		X	X	X	X	Lets you make minor corrections in any disk file either by typing the information in, or by using an existing ASCII file.
PAUSE		X	X			Pause execution for operator action. Used primarily in DO files. TRSDOS 6.0 and LDOS 5.1.3 provide this facility through JCL.
PR/FLT				X		PR/FLT is provided to format the data sent to the line printer.
PRINT	X					List a text file to the printer.
PROT	X	X	X			Use or change a diskette's master password.
PURGE		X	X	X	X	Allows you to quickly delete files from a disk.
RDUBL				X		A Disk driver program for use with Model I and the Radio Shack double density board.
RELO		X	X			Change where a program loads in memory.
REMOVE					X	Deletes a specified file or files from the disk directory, and frees the disk space for other uses. REMOVE can also be used to remove user created devices from the system. Called KILL in previous versions of TRSDOS.
RENAME	X	X	X	X	X	Changes the name of a file or device.
REPAIR				X	X	Updates and modifies information on Model I TRSDOS diskettes to make them readable under TRSDOS 6.0. Use CONV for Model III TRSDOS diskettes.
RESET				X	X	Returns a device to its original start-up condition. Closes open files if given a file specification.
ROUTE				X	X	Routes one device to another device, disk file, or to nothing (NIL.)
RS232x /DVR				X	X	This driver will configure the Model I RS-232 interface (RS232R) or the Model III RS-232 interface (RS232T). COMM/DVR in TRSDOS 6.0.
RUN				X	X	Loads and executes a machine language file. The keyword RUN is optional.
SET				X	X	Sets a driver or filter program to a device.
SETCOM		X	X		X	Adjusts the parameter values of the RS-232 port.
SETKI					X	Sets keyboard repeat rate and delay parameters.
SPOOL		X		X	X	Establishes a first-in, first-out (FIFO) buffer for a specified device (usually a line printer.) With TRSDOS 6.0 SPOOL can utilize one of the two extra 32K memory banks.
SYSGEN					X	Creates a configuration file to store information about how your system is setup.
SYSTEM				X	X	Configures certain areas of your TRSDOS system to meet your particular needs. You can set or change the disk drive configurations, and turn on or off various keyboard, video and hardware drivers.
TAPE		X	X			Used for tape to disk transfer of machine language programs.
TAPE100					X	Lets TRSDOS 6.0 read a Model 100 cassette tape file and write it to Model 4's disk, as an ASCII file or to read a disk file and write it as a Model 100 tape file.
TAPEDISK	X					Copy a machine language tape to disk.
TEST1	X					Test ROM and RAM memory.
TEST2/ BAS	X					Stress test for entire Model I system.
TIME	X	X	X	X	X	Lets you reset or display the current time.
TRACE	X	X		X	X	Displays the user's program counter address on the display. SYSTEM (TRACE) in TRSDOS 6.0.
ULC		X				Upper/lower case drive for Model Is with the Radio Shack lower case kit.
UNKILL		X				Recover a KILLed (but not ERASEd) disk file.
USER		X				Define a user library command.
VERIFY	X	X		X	X	Turns the verify function on or off. When verify is on, TRSDOS reads the data it writes to disk to verify that the data is readable.
WP		X	X	X	X	Software write-protect a diskette. Use SYSTEM under TRSDOS 6.0 and LDOS 5.1.3.
XFERSYS			X			Transfer system files in an operating system upgrade.

Maillist

Tom Kilbride
4117 Lyle
Waco, TX 76710

The most useful program you've ever published (to me, anyway) was the machine language sort in the July 1980 issue.

Having suffered through sorts of 80-100 items that take 7-8 minutes in BASIC, the program was a marvel.

I am submitting a mailing list program that uses the sort. It is written for a 48K Model I or III with disks. I have put name, address, city, state, and ZIP code into a 71-character long string. To allow for both alphabetical and ZIP code sorts, there is a subroutine that moves the ZIP code from the right hand side to the left, the sort is performed, and the ZIP is restored to the right hand side of the string.

An alpha sort of 123 names was done in less than a second. The same items sorted by ZIP took about 23 seconds with the manipulation in BASIC. Still infinitely faster than a full sort in BASIC.

After loading and running the program, the title appears on the screen while the sorting program is poked into memory. Then a filename is requested. Pressing [ENTER] defaults to a request for the number of items to be sorted. If the number is unknown, but around 100-120, 150 could be entered.

After inputting the last record, entering "END" when asked for another "Last Name?" will move you to the sorting process. The program asks whether you want to use the "Printer?" and whether you want a "ZIP Code Sort?" Then the sort begins.

At the end, you are asked whether you want to repeat the process, save the list to disk, make a correction, add a record, delete a record, or quit the program.

During the sort, the computer locks up at times. This is normal as it does a little house cleaning in the reserved string space in memory.

I estimate that you could put as many as 450 files in the program although I have not tried anything larger than 125. It probably would work best with lists of 250 or less.

After editing the program and putting in all the neat lower case lettering, I found that the sorting program and the lowercase driver have a memory conflict and the program locks up on a Model I. Since lowercase isn't essential to printing mailing labels, I'm not concerned so I'll leave it to someone else to relocate one or the other.

I have used the sorting program to alphabetize lists of names and titles of cassettes, books, etc. but I thought your readers could use this program for short mailing lists.

I offer it for publication, no strings attached. I make no claim of copyright and freely, and in the spirit of an enthusiastic TRS-80 owner hope that numerous others will be able to find application for this program or a variation of it.

For the benefit of those who were not around, let me suggest that anyone who does not have the July '80 issue rush to their nearest Radio Shack store and get the reprint of all issues (26-2115) prior to 1981. At $4.50, it's a bargain.

My TRS-80 came into this world as a 4K Level I @ $599 (before the price came down). I now have two drives, 48K, lower case mod., and a modem. I've also purchased a Model III that I use at the office and have a pocket computer. Later this year, I'm going to get a Color TRS-80 "for my kids."

Enjoy every issue so keep the good stuff coming.

```
1Ø POKE 16562, 254
     : POKE 16561, 253
2Ø CLS
     : PRINT CHR$(23)
     : PRINT @4ØØ, "MAILING LIST"
     : CLEAR 3ØØØØ
     : DEFUSRØ=&HFFØØ
     : DEFINTA-Z
     : DIMA$(45Ø), X(2)
     : Z=Ø
     : L$="XXXX/DAT"
     : GOSUB 37Ø
     : GOTO 54Ø
3Ø CLS
     : INPUT"Number on list? type <END> when done.";
     N
     : IF N>45Ø PRINT "MAX 45Ø"
     : FOR TI=Ø TO 45Ø
     : NEXT TI
     : GOTO 3Ø
4Ø FOR I=Ø TO N-1
     : GOSUB 119Ø
     : NEXT I
5Ø PRINT "Printer?"
     : GOSUB 131Ø
6Ø IF I$="Y" THEN P=1 ELSE P=Ø
7Ø PRINT "ZIP Code Sort?"
     : GOSUB 131Ø
8Ø CLS
     : IF I$="Y" THEN GOSUB 125Ø
     : FL=99 ELSE FL=Ø
9Ø X(Ø)=N
1ØØ X(1)=VARPTR(A$(Ø))
11Ø Z=USRØ(VARPTR(X(Ø)))
12Ø IF FL=99 THEN GOSUB 126Ø
13Ø FOR I=Ø TO N-1
14Ø PRINT USING"(###)"; I+1
     : GOSUB 12ØØ
15Ø NEXT I
16Ø PRINT "<R>epeat <S>ave <C>orrect <A>dd <D>elete
     <Q>uit"
     : GOSUB 131Ø
17Ø IF I$="Q" PRINT "Type 'GOTO 16Ø' to return to
     program"
     : END
18Ø IF I$="S" THEN 45Ø
19Ø IF I$="C" THEN 93Ø
2ØØ IF I$="A" THEN 64Ø
21Ø IF I$="D" THEN 127Ø
22Ø IF I$="R"CLS
     : GOTO 5Ø
23Ø GOTO 16Ø
24Ø DATA 2Ø5, 127, 1Ø, 94, 35, 86, 237, 83, 19, 255,
     35, 94, 35, 86, 237, 83
25Ø DATA 213, 255, 33, Ø, Ø, 34, 211, 255, 237, 91,
     211, 255, 2Ø3, 59, 175
26Ø DATA 2Ø3, 58, 48, 2, 2Ø3, 251, 237, 83, 211, 255,
     122, 179, 2ØØ, 42, 19
27Ø DATA 255, 237, 82, 34, 2Ø7, 255, 33, Ø, Ø, 34,
     2Ø5, 255, 42, 2Ø5, 255, 34
28Ø DATA 2Ø3, 255, 42, 2Ø3, 255, 237, 91, 211, 255,
     25, 34, 2Ø9, 255, 235, 33
29Ø DATA Ø, Ø, 25, 25, 25, 229, 237, 91, 2Ø3, 255,
     33, Ø, Ø, 25, 25, 25, 237
3ØØ DATA 75, 213, 255, 9, 235, 225, 9, 229, 213, 14,
     Ø, 126, 71, 26, 184, 48
31Ø DATA 3, 14, 1, 71, 175, 176, 4Ø, 25, 197, 19, 35,
     78, 35, 7Ø, 197, 225
32Ø DATA 235, 78, 35, 7Ø, 197, 225, 193, 26, 15Ø, 56,
     1Ø, 32, 39, 19, 35, 16
33Ø DATA 246, 2Ø3, 65, 32, 31, 2Ø9, 225, 6, 3, 78,
     235, 126, 113, 235, 119
```

```
34Ø DATA 35, 19, 16, 246, 42, 211, 255, 235, 42, 2Ø3,
     255, 175, 237, 82, 34
35Ø DATA 2Ø3, 255, 48, 144, 24, 2, 2Ø9, 225, 42, 2Ø5,
     255, 17, 1, Ø, 175, 25
36Ø DATA 34, 2Ø5, 255, 237, 91, 2Ø7, 255, 237, 82,
     218, 58, 255, 195, 24, 255
37Ø N=Ø
38Ø FOR I=1 TO 2Ø3
39Ø READ A
4ØØ N=N+A
41Ø POKE I-257, A
42Ø NEXT
43Ø IF N<>25337 THEN PRINT "C R A S H ! ! !"
     : END
44Ø RETURN
45Ø PRINT "Use "; LEFT$(L$, LEN(L$)-4); "?"
     : GOSUB 131Ø
     : IF I$="Y" THEN 47Ø ELSE INPUT "FILENAME"; L$
     : IF LEN(L$)>8 GOTO 45Ø
46Ø L$=L$+"/DAT"
47Ø OPEN "O", 1, L$
48Ø PRINT #1, N
49Ø FOR I=Ø TO N-1
5ØØ PRINT #1, A$(I)
51Ø NEXT I
52Ø CLOSE
53Ø GOTO 16Ø
54Ø CLS
     : INPUT "Filename"; L$
     : IF LEN(L$)>8 THEN L$="XXXX/DAT"
     : GOTO 54Ø
55Ø IF L$="XXXX/DAT" THEN 3Ø  ELSE L$=L$+"/DAT"
56Ø OPEN "I", 1, L$
57Ø INPUT #1, N
58Ø FOR I=Ø TO N-1
59Ø INPUT #1, A$(I)
6ØØ NEXT I
61Ø CLOSE
62Ø PRINT "Add Data?"
     : GOSUB 131Ø
63Ø IF I$="N" THEN 5Ø
64Ø CLS
     : PRINT "This is Item -"; N+1
65Ø GOSUB 66Ø
     : GOSUB 71Ø
     : GOSUB 75Ø
     : GOSUB 79Ø
     : GOSUB 83Ø
     : GOSUB 86Ø
     : GOTO 9ØØ
66Ø INPUT "Last Name >"; I$
     : N$=I$
     : I$=""
67Ø IF LEN(N$)>15 PRINT "15 MAX"
     : GOTO 66Ø
68Ø IF N$="END" OR N$="end" THEN 5Ø
69Ø N$=N$+STRING$(15-LEN(N$), 32)
7ØØ RETURN
71Ø INPUT "First Name >"; I$
     : F$=I$
     : I$=""
72Ø IF LEN(F$)>1Ø PRINT "1Ø MAX"
     : GOTO 71Ø
73Ø F$=F$+STRING$(1Ø-LEN(F$), 32)
74Ø RETURN
75Ø INPUT "Street Address >"; I$
     : A$=I$
     : I$=""
76Ø IF LEN(A$)>2Ø PRINT "2Ø MAX"
     : GOTO 75Ø
77Ø A$=A$+STRING$(2Ø-LEN(A$), 32)
78Ø RETURN
79Ø INPUT "City >"; I$
     : C$=I$
     : I$=""
8ØØ IF LEN(C$)>15 PRINT "15 MAX"
     : GOTO 79Ø
81Ø C$=C$+STRING$(15-LEN(C$), 32)
82Ø RETURN
83Ø INPUT "State >"; I$
     : S$=I$
     : I$=""
84Ø IF LEN(S$)<>2 PRINT "2 ONLY"
     : GOTO 83Ø
85Ø RETURN
86Ø INPUT "ZIP >"; I$
     : Z$=I$
     : I$=""
87Ø IF LEN(Z$)>9 PRINT "9 MAX"
     : GOTO 86Ø
88Ø Z$=Z$+STRING$(9-LEN(Z$), 32)
89Ø RETURN
9ØØ A$(N)=N$+F$+A$+C$+S$+Z$
91Ø N=N+1
92Ø CLS
     : GOTO 64Ø
93Ø CLS
     : INPUT "No. to Correct"; NN
94Ø IF NN<1 OR NN>N THEN 93Ø
95Ø CLS
     : PRINT @128, "Item #"; NN
     : PRINT
     : IF LEN(A$(NN-1))<71 THEN FOR X=1 TO
     71-LEN(A$(NN-1))
     : A$(NN-1)=A$(NN-1)+" "
     : NEXT X ELSE A$(NN-1)=LEFT$(A$(NN-1), 71)
96Ø X$=A$(NN-1)
     : PRINT "1 - Last Name >"; LEFT$(X$, 15)
97Ø PRINT "2 - First Name >"; MID$(X$, 16, 1Ø)
     : PRINT "3 - Address >"; MID$(X$, 26, 2Ø)
98Ø PRINT "4 - City >"; MID$(X$, 46, 15)
     : PRINT "5 - State >"; MID$(X$, 61, 2)
     : PRINT "6 - ZIP Code >"; RIGHT$(X$, 9)
     : PRINT "7 - No Change"
99Ø PRINT
     : PRINT "Change What Item?"
1ØØØ GOSUB 131Ø
     : B=VAL(I$)
     : I$=""
     : IF B=Ø THEN 1ØØØ
1Ø1Ø ON B GOTO 1Ø2Ø ,1Ø4Ø ,1Ø6Ø ,1Ø8Ø ,11ØØ ,113Ø,
     116Ø ,116Ø ,116Ø ,116Ø
1Ø2Ø GOSUB 66Ø
1Ø3Ø X$=N$+RIGHT$(X$, 56)
     : GOTO 115Ø
1Ø4Ø GOSUB 71Ø
1Ø5Ø X$=LEFT$(X$, 15)+F$+RIGHT$(X$, 46)
     : GOTO 115Ø
1Ø6Ø GOSUB 75Ø
1Ø7Ø X$=LEFT$(X$, 25)+A$+RIGHT$(X$, 26)
     : GOTO 115Ø
1Ø8Ø GOSUB 79Ø
1Ø9Ø X$=LEFT$(X$, 45)+C$+RIGHT$(X$, 11)
     : GOTO 115Ø
11ØØ GOSUB 83Ø
111Ø X$=LEFT$(X$, 6Ø)+S$+RIGHT$(X$, 9)
112Ø GOTO 115Ø
113Ø GOSUB 86Ø
114Ø X$=LEFT$(X$, 62)+Z$
115Ø A$(NN-1)=X$
     : GOTO 95Ø
116Ø A$(NN-1)=X$
     : X$=""
     : GOTO 16Ø
117Ø CLS
118Ø GOTO 16Ø
119Ø CLS
     : PRINT "Item #"; I+1
     : GOSUB 66Ø
     : GOSUB 71Ø
     : GOSUB 75Ø
     : GOSUB 79Ø
     : GOSUB 83Ø
     : GOSUB 86Ø
```

```
     : A$(I)=N$+F$+A$+C$+S$+Z$
     : RETURN
12ØØ N$=LEFT$(A$(I), 15)
     : F$=MID$(A$(I), 16, 1Ø)
     : A$=MID$(A$(I), 26, 2Ø)
     : C$=MID$(A$(I), 46, 15)
     : S$=MID$(A$(I), 61, 2)
     : Z$=RIGHT$(A$(I), 9)
     : IF RIGHT$(Z$, 4)<>"   "Z$=LEFT$(Z$,
     5)+"-"+RIGHT$(Z$, 4)
121Ø PRINT N$+" "+F$
     : PRINT A$
     : PRINT C$+" "+S$+" "+Z$
122Ø IF P<>1 THEN RETURN
123Ø FOR X=1 TO 1Ø
     : IF MID$(F$, X, 2)<>"  " THEN NEXT X ELSE
     F$=LEFT$(F$, X)
124Ø LPRINT F$+N$
     : LPRINT A$
     : LPRINT C$+" "+S$+""+Z$
     : LPRINT
     : LPRINT
     : LPRINT
     : RETURN
125Ø FOR I=Ø TO N-1
     : A$(I)=RIGHT$(A$(I), 9)+LEFT$(A$(I),
     LEN(A$(I))-9)
     : NEXT I
     : RETURN
126Ø FOR I=Ø TO N-1
     : A$(I)=RIGHT$(A$(I), LEN(A$(I))-9)+
     LEFT$(A$(I), 9)
     : NEXT I
     : RETURN
127Ø CLS
     : INPUT "Delete No."; NN
     : IF NN<1 OR NN>N-1 THEN 127Ø
128Ø PRINT LEFT$(A$(NN-1), 25)
     : PRINT RIGHT$(A$(NN-1), 46)
     : PRINT "Delete This?"
129Ø GOSUB 131Ø
     : IF I$="Y" THEN 13ØØ ELSE IF I$="N" THEN 16Ø
     ELSE 129Ø
13ØØ FOR I=NN-1 TO N-1
     : A$(I)=A$(I+1)
     : NEXT I
     : N=N-1
     : GOTO 16Ø
131Ø I$=INKEY$
     : IF I$="" THEN 131Ø
132Ø IF ASC(I$)>96 AND ASC(I$)<123 THEN I$=
     CHR$(ASC(I$)-32)
133Ø RETURN
```

Fast Multiply

James Evans
38 Belaire Drive
Stamford, CT 06905

I have written a program for the Model III that you might be interested in, fast double precision multiply. By use of the alternate registers and some explicit straight line coding, I was able to reduce the execution time from some 21 ms, (+6 ms BASIC overhead time when used in a FOR loop) to 6 ms (+ the same BASIC overhead, unfortunately) the method of diversion and testing at RST20 allows this program to be used by existing BASIC programs without change (except memory must be protected above 64,000—where this program resides).

Here is the BASIC program and the first page of the assembly listing which shows the dimension principle involved.

I have in the past year written an entire library of routines like this.

Program 1. Assembly Language

```
FAØØ             ØØ1ØØ              ORG       ØFAØØH
FAØØ  F3         ØØ11Ø  DPSW                  DI
FAØ1  Ø8         ØØ12Ø              EX        AF,AF'
FAØ2  AF         ØØ13Ø              XOR       A
FAØ3  3EØØ       ØØ14Ø              LD        A,Ø
FAØ5  32ØA4Ø     ØØ15Ø              LD        (4ØØAH),A
FAØ8  3EFA       ØØ16Ø              LD        A,ØFAH
FAØA  32ØB4Ø     ØØ17Ø              LD        (4ØØBH),A
FAØD  D9         ØØ18Ø              EXX
FAØE  C1         ØØ19Ø              POP       BC
FAØF  D1         ØØ2ØØ              POP       DE
FA1Ø  21DCØA     ØØ21Ø              LD        HL,ØADCH
FA13  ED42       ØØ22Ø              SBC       HL,BC
FA15  C23AFA     ØØ23Ø              JP        NZ,RESTOR
FA18  2i4E24     ØØ24Ø              LD        HL,244EH
FA1B  ED52       ØØ25Ø              SBC       HL,DE
FA1D  C23AFA     ØØ26Ø              JP        NZ,RESTOR
FA2Ø  3ABØ4Ø     ØØ27Ø              LD        A,(4ØBØH)
FA23  FEØ2       ØØ28Ø              CP        2
FA25  C23AFA     ØØ29Ø              JP        NZ,RESTOR
FA28  Ø8         ØØ3ØØ              EX        AF,AF'
FA29  D9         ØØ31Ø              EXX
FA2A  CDD925     ØØ32Ø              CALL      25D9H
FA2D  CDDCØA     ØØ33Ø              CALL      ØADCH
FA3Ø  CD42FA     ØØ34Ø              CALL      MDP
FA33  CD8AFD     ØØ35Ø              CALL      RENOR
FA36  FB         ØØ36Ø              EI
FA37  C9         ØØ37Ø              RET
FA38  C9         ØØ38Ø              RET
FA39  C9         ØØ39Ø              RET
FA3A  D5         ØØ4ØØ  RESTOR      PUSH      DE
FA3B  C5         ØØ41Ø              PUSH      BC
FA3C  Ø8         ØØ42Ø              EX        AF,AF'
FA3D  D9         ØØ43Ø              EXX
FA3E  C3D925     ØØ44Ø  FIN         JP        25D9H
FA41  ØØ         ØØ45Ø  MULTDP      NOP
ØØ4Ø             ØØ46Ø  S           EQU       4ØH
FR42  F3         ØØ47Ø  MDP         DI
FA43  211D41     ØØ48Ø              LD        HL,411DH
FA46  Ø1Ø6ØØ     ØØ49Ø              LD        BC,6H
FE4Ø             ØØ5ØØ  START       EQU       ØFE4ØH
FA49  114ØFE     ØØ51Ø              LD        DE,START
FA4C  EDBØ       ØØ52Ø              LDIR
FA4E  7E         ØØ53Ø              LD        A,(HL)
FA4F  F68Ø       ØØ54Ø              OR        SØH
FA51  12         ØØ55Ø              LD        (DE),A
FA52  1C         ØØ56Ø              INC       E
FA53  3EØØ       ØØ57Ø              LD        A,Ø
FA55  12         ØØ58Ø              LD        (DE),A
FA56  ØØ         ØØ59Ø              NOP
FA57  212741     ØØ6ØØ              LD        HL,4127H
FA5A  Ø1Ø6ØØ     ØØ61Ø              LD        BC,6H
FA5D  1138FE     ØØ62Ø              LD        DE,START-8
FA6Ø  EDBØ       ØØ63Ø              LDIR
FA62  7E         ØØ64Ø              LD        A,(HL)
FA63  F68Ø       ØØ65Ø              OR        8ØH
FA65  12         ØØ66Ø              LD        (DE),A
FA66  1C         ØØ67Ø              INC       E
FA67  3EØØ       ØØ68Ø              LD        A,Ø
FA69  12         ØØ69Ø              LD        (DE),A
FA6A  214ØFE     ØØ7ØØ              LD        HL,START
```

Program 2. BASIC Program

```
1ØØ 'REQUIRES PROTECTED MEMORY FOR FAØØ
11Ø 'SWITCH DETECTS THE PRESENCE OF DOUBLE PRECISION
    ARITHMETIC
```

```
12Ø 'COMMAND BY THE NUMBER TYPE TEST IN RESET
    2Ø,WHICH IS
13Ø 'MADE JUST BEFORE THE ACTUAL OPERATION
    DETERMINATION.SINCE
14Ø 'MANY ROM ROUTINES SO TEST,THE UNIQUE
    DETERMINATION AS
15Ø 'TO WHERE IT CAME FROM IS MADE BY TESTING THE
    LAST TWO
16Ø 'ENTRIES ON THE STACK.THE DRIVER CODE FOR THE
    ARITHMETIC
17Ø 'OPERATIONS IS MADE AT CODE STARTING AT 24Ø6
    XMATH/SRC
18Ø 'THE TYPE OF ARITHMETIC IS AT 4ØBØ
    : MULTIPLY IS 2,
19Ø 'TESTING FOR 2 MEANS ONLY MULT IS DIVERTED
2Ø5 CMD"L","DPSW/CMD"
21Ø DEFUSRØ=&HFAØØ
22Ø DEFDBLA-K
23Ø X=USR(Ø)
24Ø A=1/7DØ
    : B=1/3DØ
25Ø C=A*B
    : PRINT C
255 STOP
26Ø DEFUSRØ=&HFE14
    : X=USR(Ø)
    : 'RESTORES RST 2Ø, ELSE REBOOT
```

Alphabet

Michael G. Lofy
2622 Moraine Court
Racine, WI 53402

I have developed a program for the Model III that utilizes some of the methods that are described in your publication, BASIC FASTER AND BETTER.

It is an entertaining program for children using high speed graphics.

```
5  ' CLEAR1ØØØ
1Ø '********************
2Ø '*  Alphabet Primer *
3Ø '*  Michael G Lofy  *
4Ø '*  May 15, 1982    *
5Ø '********************
6Ø '** This program responds to the push of a letter
    by drawing
7Ø '** a picture in high speed graphics  It responds
    to a
8Ø '** number by duplicating the picture that many
    times.
9Ø '** Zero is ten.  This program uses techniques
    from BASIC
91 '** FASTER AND BETTER by Lewis Rosenfelder.  Copy
    the
92 '** program EXACTLY.  Comments and spaces may be
    deleted
93 '** to speed execution.  DO NOT USE ANY OTHER
    VARIABLES
94 '** UNLESS THEY ARE DEFINED PRIOR TO LINE 11Ø.
1ØØ X%=Ø
    : X1%=Ø
    : Z%=Ø
    : MT%=Ø
    : MF%=Ø
    : J%=Ø
    : XN%=Ø
    : XH%=Ø
    : IN$=""
    : X2%=Ø
1Ø9 '** Store USR subroutine in "MAGIC ARRAY"
11Ø DIM US%(7)
12Ø US%(Ø)=8448
    : US%(2)=4352
    : US%(4)=256
    : US%(7)=2Ø1
    : US%(6)=-2Ø243
129 '** Dimension integer array to store coded
    pictures
13Ø DIM A%(1Ø92)
14Ø DEFUSRØ=VARPTR(US%(Ø))
148 '** Set space compression software switch off to
    print
149 '** special characters on MODEL III
15Ø PRINT CHR$(21)
    : CLS
16Ø PRINT "PLEASE WAIT"
168 '** Read DATA into array. This array stores all
    the pictures
169 '** in coded format for later relocation to
    screen.
17Ø FOR X%=1 TO 1Ø92
18Ø READ A%(X%)
22Ø NEXT X%
    : CLS
228 '** Pressing a letter key will draw a picture.
    All other
229 '** keys are locked out.
23Ø PRINT @ 96Ø, "PRESS A LETTER OF THE ALPHABET";
24Ø IN$=""
    : IN$=INKEY$
    : IF IN$="" THEN 24Ø
25Ø IF ASC(IN$)<65 OR ASC(IN$)>9Ø THEN 24Ø
26Ø MF%=((ASC(IN$)-65)*42)+1
267 '** USR subroutine relocates information.  Based
    on the
268 '** number  pressed, the coded information is
    transferred
269 '** from the array and POKED into screen memory.
27Ø CLS
28Ø MT%=15360-64
29Ø FOR X%=MF% TO MF%+41 STEP 6
3ØØ MT%=MT%+64
31Ø US%(5)=12
32Ø US%(1)=VARPTR(A%(X%))
33Ø US%(3)=MT%
34Ø J%=USRØ(Ø)
35Ø NEXT X%
356 '** Pressing a number key will duplicate the
    picture
357 '** that many times.  Zero equals ten.
36Ø PRINT @ 96Ø, "PRESS A NUMBER";
37Ø IN$=INKEY$
    : IF IN$="" THEN 37Ø
38Ø IF ASC(IN$)<48 OR ASC(IN$)>57 THEN 37Ø
39Ø XN%=VAL(IN$)
4ØØ IF XN%=Ø THEN XN%=1Ø
41Ø XH%=(XN%/2)-1
42Ø IF XN%=1 THEN 23Ø
43Ø IF XN%=2 THEN GOSUB 5Ø13Ø
    : GOTO 23Ø
44Ø IF XN%=3 THEN GOSUB 5Ø13Ø
    : XH%=1
    : GOSUB 5Ø19Ø
    : GOTO 23Ø
45Ø GOSUB 5ØØ1Ø
46Ø GOSUB 5Ø13Ø
47Ø IF XN% AND NOT-2 THEN XH%=XH%+1
    : GOSUB 5Ø19Ø
48Ø GOTO 23Ø
5ØØØØ '*** Subroutine to duplicate to the side.
5ØØ1Ø FOR X%=1 TO XH%
5ØØ2Ø MT%=15296+(X%*13)
5ØØ3Ø FOR X1%=Ø TO 6
5ØØ4Ø MT%=MT%+64
5ØØ5Ø US%(1)=1536Ø+(X1%*64)
5ØØ6Ø US%(3)=MT%
5ØØ7Ø US%(5)=12
```

```
5ØØ8Ø J%=USRØ(Ø)
5ØØ9Ø NEXT X1%
5Ø1ØØ NEXT X%
5Ø11Ø RETURN
5Ø12Ø '*** This subroutine duplicates the top half to
     the bottom
5Ø13Ø US%(1)=1536Ø
5Ø14Ø US%(3)=15872
5Ø15Ø US%(5)=512
5Ø16Ø J%=USRØ(Ø)
5Ø17Ø RETURN
5Ø18Ø '*** This subroutine takes care of odd numbers
5Ø19Ø MT%=15296+(XH%*13)
5Ø2ØØ FOR X%=Ø TO 6
5Ø21Ø MT%=MT%+64
5Ø22Ø US%(1)=1536Ø+(X%*64)
5Ø23Ø US%(3)=MT%
5Ø24Ø US%(5)=12
5Ø25Ø J%=USRØ(Ø)
5Ø26Ø NEXT X%
5Ø27Ø RETURN
5998Ø '** The data below loads the integer array with
     coded
59981 '** information describing the pictures.  By
     PEEKing
59982 '** VARPTR(A%(x)) and VARPTR(A%(x)+1), you will
     see
59983 '** the CHR$ codes that are POKEd to the screen
     with
59984 '** USR subroutine.
59999 '** ARROW
6ØØØØ DATA  8224, 8224, -184ØØ, -28483, 8224, 8224
6ØØ1Ø DATA  8224, -24544, -1645Ø, -16449, 8372, 8224
6ØØ2Ø DATA  8224, -31968, -21629, -318Ø9, -32381,
     8224
6ØØ3Ø DATA  8224, 8224, -21984, 8383, 8224, 8224
6ØØ4Ø DATA  8224, 8224, -21984, 8383, 8224, 8224
6ØØ5Ø DATA  8224, 8224, -21984, 8383, 8224, 8224
6ØØ6Ø DATA  21Ø57, 2Ø3Ø6, 8279, 8224, 8224, 8224
6ØØ69 '** BLOCK
6ØØ7Ø DATA  8224, -29536, -29556, -29556, -21364,
     834Ø
6ØØ8Ø DATA -26592, 8321, 8224, 8224, -32368Ø, 8341
6ØØ9Ø DATA -29538, -29556, -29556, -26996, 8224, 8341

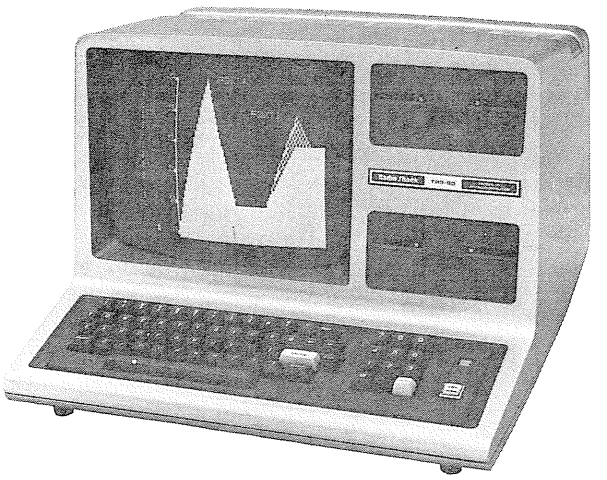
6Ø1ØØ DATA  8341, 8224, 8224, -27368Ø, -31Ø72, 8224
6Ø11Ø DATA -2Ø299, -2Ø3Ø4, -2Ø3Ø4, -25168, 8321, 8224

6Ø12Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø13Ø DATA  19522, 17231, 8267, 8224, 8224, 8224
6Ø139 '** CARD
6Ø14Ø DATA -318Ø9, -31869, -31869, -31869, -31869,
     -165Ø9
6Ø15Ø DATA  16831, 8386, 8224, 8336, 8224, -166Ø8
6Ø16Ø DATA  8383, 8224, -16736, -19265, 8224, -166Ø8
6Ø17Ø DATA  8383, 8224, -2Ø6Ø6, -3Ø785, 8224, -166Ø8
6Ø18Ø DATA  8383, 8224, 8224, 8321, 16672, -16446
6Ø19Ø DATA -2Ø289, -2Ø3Ø4, -2Ø3Ø4, -2Ø3Ø4, -2Ø3Ø4,
     -16464
6Ø2ØØ DATA  167Ø7, 1749Ø, 8224, 8224, 8224, 8224
6Ø2Ø9 '** DIAMOND
6Ø21Ø DATA  8224, 8224, -16736, -28483, 8224, 8224
6Ø22Ø DATA  8224, -184ØØ, -16449, -16449, 8372, 8224
6Ø23Ø DATA -24544, -1645Ø, -16449, -16449, -16961,
     8336
6Ø24Ø DATA -32224, -16465, -16449, -16449, -24641,
     8321
6Ø25Ø DATA  8224, -2992Ø, -16449, -16449, 8327, 8224
6Ø26Ø DATA  8224, 8224, -2Ø6Ø6, -32353, 8224, 8224
6Ø27Ø DATA  18756, 19777, 2ØØ47, 826Ø, 8224, 8224
6Ø279 '** ENVELOPE
6Ø28Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø29Ø DATA -2Ø3Ø4, -2Ø3Ø4, -2Ø3Ø4, -2Ø3Ø4, -2Ø3Ø4,
     -2Ø3Ø4
6Ø3ØØ DATA  18Ø69, 2Ø3Ø6, 8269, 8224, -27368Ø, -21763
6Ø31Ø DATA  8341, 8224, 2Ø3Ø8, 8224, -31968, -21629
6Ø32Ø DATA -31869, -31869, -31869, -31869, -31869,
     -31869
6Ø33Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø34Ø DATA  2ØØ37, 17758Ø, 2Ø3ØØ, 17744, 8224, 8224
6Ø349 '** FISH
6Ø35Ø DATA  8224, 8224, -17248, -1722Ø, 8372, 8224
6Ø36Ø DATA  8224, -184ØØ, -16449, -25665, -16961,
     8336
6Ø37Ø DATA -2Ø291, -1645Ø, -16449, -24641, -17222,
     -19265
6Ø38Ø DATA -31841, -16465, -16449, -167Ø5, -16449,
     -3Ø785
6Ø39Ø DATA  8224, -2992Ø, -16449, -16449, -24641,
     8321
6Ø4ØØ DATA  8224, 8224, -28798, -28785, 8327, 8224
6Ø41Ø DATA  18758, 18515, 8224, 8224, 8224, 8224
6Ø419 '** GUN
6Ø42Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø43Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø44Ø DATA -16449, -16449, -16449, -16449, -16449,
     -2958Ø5
6Ø45Ø DATA -16449, -3Ø273, -24432, 8325, 8224, 8224
6Ø46Ø DATA -16449, -318Ø9, -32381, 8224, 8224, 8224
6Ø47Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø48Ø DATA  21831, 827Ø, 8224, 8224, 8224, 8224
6Ø489 '** HAT
6Ø49Ø DATA  8224, 8224, 8224, 8224, 8224, 8224
6Ø5ØØ DATA  8224, -16449, -16449, -16449, -16449,
     8224
6Ø51Ø DATA  8224, -16449, -16449, -16449, -16449,
     8224
6Ø52Ø DATA  8224, -16449, -16449, -16449, -16449,
     8224
6Ø53Ø DATA  8224, -16449, -16449, -16449, -16449,
     8224
6Ø54Ø DATA -2Ø32Ø, -16449, -16449, -16449, -16449,
     -28496
6Ø55Ø DATA  16712, 8276, 8224, 8224, 8224, 8224
6Ø559 '** ICE CREAM
6Ø56Ø DATA -184ØØ, -1645Ø, -16449, -16449, -16961,
     8372
6Ø57Ø DATA -2992Ø, -16449, -16449, -16449, -16449,
     8327
6Ø58Ø DATA  8224, -23264, 8224, 8224, 8346, 8224
6Ø59Ø DATA  8224, -32224, 834Ø, -22496, 8321, 8224
6Ø6ØØ DATA  8224, 8224, -28534, -31328, 8224, 8224
6Ø61Ø DATA  8224, 8224, -2224Ø, 8342, 8224, 8224
6Ø62Ø DATA  17225, 8261, 21Ø59, 167Ø9, 8269, 8224
6Ø629 '** JET
6Ø63Ø DATA  8224, 8224, -16864, 8381, 8224, 8224
6Ø64Ø DATA  8224, 8224, -16456, -19265, 8224, 8224
6Ø65Ø DATA  8224, -16736, -16481, -2Ø545, -28483,
     8224
6Ø66Ø DATA -184ØØ, -3Ø785, -166Ø8, 8383, -165Ø1, 8372
6Ø67Ø DATA -31968, 8321, -166Ø8, 8383, -32224, 8323
6Ø68Ø DATA  8224, -24544, -16482, -21Ø57, 8336, 8224
6Ø69Ø DATA  17738, 8276, 8224, 8224, 8224, 8224
6Ø699 '** KEY
6Ø7ØØ DATA  8224, 8224, -22496, -1722Ø, -31556, 8224
6Ø71Ø DATA  8224, 8224, -21984, 8341, -32381, 8224
6Ø72Ø DATA  8224, 8224, -21984, 8341, 8224, 8224
6Ø73Ø DATA  8224, 8224, -17888, 8373, 8224, 8224
6Ø74Ø DATA  8224, -24544, -32354, -21118, 8336, 8224
6Ø75Ø DATA  8224, 8224, -19317, -3Ø792, 8224, 8224
6Ø76Ø DATA  17739, 8281, 8224, 8224, 8224, 8224
6Ø769 '** LETTER
6Ø77Ø DATA -2954Ø, -29556, -29556, -29556, -29556,
     -21364
6Ø78Ø DATA  17557, 167Ø9, 11346, 8224, 8224, -21984
6Ø79Ø DATA  8341, 17736, 19532, 8271, 8224, -21984
6Ø8ØØ DATA  8341, 8224, 8224, 8224, 8224, -21984
6Ø81Ø DATA  8341, 8224, 8224, 2Ø3ØØ, 17758Ø, -21972
6Ø82Ø DATA -29555, -29556, -29556, -29556, -29556,
     -29Ø44
6Ø83Ø DATA  1774Ø, 21588, 21Ø61, 8224, 8224, 8224
6Ø839 '** MONEY
6Ø84Ø DATA -2954Ø, -29556, -29556, -29556, -29556,
     8364
```

```
60850 DATA  8341, 9248, 11825, 12336, 8224, 8362
60860 DATA -29555, -29556, -29556, -29556, -29556,
      8334
60870 DATA -25568, -29556, -29556, -29556, -29556,
      -21364
60880 DATA -27360, 8224, 13604, 12334, 8240, -21984
60890 DATA -29408, -29556, -29556, -29556, -29556,
      -29044
60900 DATA  20301, 17742, 8281, 8224, 8224, 8224
60909 '** NOTE
60910 DATA  8224, 8224, -22496, 8336, 8224, 8224
60920 DATA  8224, 8224, -21984, -19317, 8224, 8224
60930 DATA  8224, 8224, -21984, -32224, 8224, 8224
60940 DATA  8224, 8224, -21984, 8224, 8224, 8224
60950 DATA -18400, -16449, -16705, 8224, 8224, 8224
60960 DATA -32224, -28785, -32369, 8224, 8224, 8224
60970 DATA  20302, 17748, 8224, 8224, 8224, 8224
60979 '** OSTRICH
60980 DATA -24544, -21328, -28484, 8224, 8224, 8224
60990 DATA  8224, -31968, 8343, 8224, 8224, 8224
61000 DATA  8224, 8224, -20331, -17220, -19268, 8224
61010 DATA  8224, -18400, -16449, -16449, -16449,
      -27457
61020 DATA  8224, -32224, -28789, -20577, -28785,
      8327
61030 DATA  8224, 8224, -26592, -32127, 8356, 8224
61040 DATA  21327, 21076, 17225, 8264, 8224, 8224
61049 '** PILLAR
61050 DATA -31968, -16497, -16449, -16449, -28737,
      8323
61060 DATA  8224, 8224, -16449, -16449, 8224, 8224
61070 DATA  8224, 8224, -16449, -16449, 8224, 8224
61080 DATA  8224, 8224, -16449, -16449, 8224, 8224
61090 DATA  8224, 8224, -16449, -16449, 8224, 8224
61100 DATA -17376, -16450, -16449, -16449, -16961,
      8380
61110 DATA  18768, 19532, 21057, 8224, 8224, 8224
61119 '** QUESTION
61120 DATA  8224, -24544, -28772, -29297, 8372, 8224
61130 DATA  8224, -18552, 8224, 8224, -31302, 8224
61140 DATA  8224, 8224, -22496, -29028, 8321, 8224
61150 DATA  8224, 8224, -21984, 8341, 8224, 8224
61160 DATA  8224, 8224, -21984, 8341, 8224, 8224
61170 DATA  8224, 8224, -30688, 8324, 8224, 8224
61180 DATA  21841, 21317, 18772, 20047, 8224, 8224
61189 '** ROBOT
61190 DATA  8224, 8224, -18256, -20300, 8224, 8224
61200 DATA  8224, 8224, -31595, -21880, 8224, 8224
61210 DATA -20448, -20304, -17485, -19529, -20304,
      8368
61220 DATA -27360, -16470, -16449, -16449, -27201,
      8362
61230 DATA -21080, -28790, -28737, -16497, -31345,
      -27490
61240 DATA  8224, -20320, 8383, -16608, -28496, 8224
61250 DATA  20306, 20290, 8276, 8224, 8224, 8224
61259 '** STAIRS
61260 DATA  8224, 8224, 8224, 8224, 8224, 8224
61270 DATA  8224, 8224, 8224, 8224, 8224, 8224
61280 DATA  8224, 8224, 8224, 8224, 8224, 8224
61290 DATA  8224, 8224, 8224, -17232, -16449, -16449
61300 DATA  8224, -20448, -16452, -16449, -16449,
      -16449
61310 DATA -17232, -16449, -16449, -16449, -16449,
      -16449
61320 DATA  21587, 18753, 21330, 8224, 8224, 8224
61329 '** TREE
61330 DATA  8224, 8224, 8224, 8224, 8224, 8224
61340 DATA  8224, -25424, -27769, -29813, -20308,
      8224
61350 DATA -30176, -28491, -30559, -28384, -17758,
      8325
61360 DATA  8224, -31968, -21363, -29028, 8323, 8224
61370 DATA  8224, 8224, -21984, 8341, 8224, 8224
61380 DATA  8224, 8224, -16720, -20291, 8224, 8224
61390 DATA  21076, 17733, 8224, 8224, 8224, 8224
61399 '** UMBRELLA
61400 DATA  8224, 8224, -24544, 8224, 8224, 8224
61410 DATA  8224, -17232, -16450, -17217, -28492,
      8224
61420 DATA -16456, -16449, -16449, -16449, -16449,
      -28483
61430 DATA -32373, -32373, -21621, -32373, -32373,
      -32373
61440 DATA  8224, 8224, -21984, 8224, 8368, 8224
61450 DATA  8224, 8224, -32224, -31095, 8321, 8224
61460 DATA  19797, 21058, 19525, 16716, 8224, 8224
61469 '** VACUUM
61470 DATA  8224, 8224, 8224, 8224, 8224, 8224
61480 DATA  8224, 8224, -25440, 8324, 8224, 8224
61490 DATA  8224, 8224, -28481, 8224, 8224, 8224
61500 DATA  8224, 8224, -16449, 8372, 8224, 8224
61510 DATA  8224, 8224, -16449, -27201, 8224, 8224
61520 DATA -18272, -16452, -28737, 8323, 8224, 8224
61530 DATA  16726, 21827, 19797, 8224, 8224, 8224
61539 '** WRENCH
61540 DATA  8224, 8224, 8368, 8368, 8224, 8224
61550 DATA  8224, 8224, -16509, 8323, 8224, 8224
61560 DATA  8224, 8224, -16608, 8224, 8224, 8224
61570 DATA  8224, 8224, -16608, 8224, 8224, 8224
61580 DATA  8224, -21984, -31849, -27221, 8224, 8224
61590 DATA  8224, 8224, 8224, 8224, 8224, 8224
61600 DATA  21079, 20037, 18499, 8224, 8224, 8224
61609 '** XYLOPHONE
61610 DATA  8224, -29540, -29556, -29556, -27508,
      8224
61620 DATA -27360, -19574, -19533, -19533, 8347, 8224

61630 DATA -27360, -22240, -29556, -21364, 8321, 8362

61640 DATA -27360, 8224, -19545, -31309, 8224, 8362
61650 DATA -28790, 8224, -21374, 8326, 8224, 8362
61660 DATA  8224, 8224, 8224, 8224, 8224, -31345
61670 DATA  22872, 20300, 18512, 20047, 8261, 8224
61679 '** YIELD SIGN
61680 DATA -27765, -31869, -31869, -31869, -31869,
      -30813
61690 DATA -32224, 22948, 17737, 17484, -26592, 8321
61700 DATA  8224, -30432, 8336, -24544, 8326, 8224
61710 DATA  8224, 8224, -23422, -32360, 8224, 8224
61720 DATA  8224, 8224, -21984, 8341, 8224, 8224
61730 DATA  8224, 8224, -21984, 8341, 8224, 8224
61740 DATA  18777, 19525, 8260, 18771, 20039, 8224
61749 '** ZIPPER
61750 DATA  8224, 8224, -19424, -27512, 8224, 8224
61760 DATA  8224, 8224, -25312, -27230, 8224, 8224
61770 DATA  8224, 8224, -18656, -27256, 8224, 8224
61780 DATA  8224, 8224, -25312, -27230, 8224, 8224
61790 DATA  8224, 8224, -18656, -27256, 8224, 8224
61800 DATA  8224, 8224, -21728, -32325, 8224, 8224
61810 DATA  18778, 20560, 21061, 8224, 8224, 8224
```



Sound Generator

Jerry W. Stajduhar
5114 Lodgepole Lane
Mountain Home AFB, ID 83648

Both tone programs were written for the Model I, but might work on the Model III if the cassette port is addressed and controlled the same (I'm not sure how the Model III works). The listings are of two BASIC programs that can make sound when called by the USR(X) function. The programs are identical, except one is for 16K non-disk and the other is for 48K disk systems.

INSTRUCTIONS FOR USE OF THE 'TONE' PROGRAM

The tone program is designed to be used on the TRS-80 Model I to give the computer the ability to generate sound under BASIC control (with the help of a small amplifier connected to the cassette aux. plug).

To use the program, first set the correct memory size (see program listing for your computer). Then enter your main program, adding in the tone program somewhere at the beginning. The tone program need only be executed once. It generates a machine language program that the computer calls by using the USR command. I usually put the tone program on the first four lines to insure that it gets executed (using the USR function without having a machine language program in memory tends to be unhealthy)! Once the data has been read into memory it is ready to generate tones.

To tell the tone program what sounds you want it to make, a 'music table' must be poked into memory (it's very easy). The music table starts at 32700 for the 16K version and −136 for the 48K disks version (−136 is BASIC's way of saying 65400). First, you must decide how long you want the tone to last. You have a choice of 1-255, with 1 being the shortest and 255 being the longest. You then poke this value at the start of the music table (32700 for 16K, −136 for 48K disk). Then you decide on what tone you want. You can have 0-255; 1 is highest; 255 is lowest; 0 is a rest (silence). You must then poke this value into the next memory location in the music table (32701 for 16K, −135 for 48K disk). Then you decide on the length of the next tone and poke that value into the next location (32702, −134), and the same with the next note. You continue to make the table in the same fashion until you are done with whatever sound you are trying to make. After you have poked the last tone into memory, you need to tell the computer that you are finished. This is done by poking 0 in for the next duration. The computer recognizes this as the end of the music table.

To get the computer to play the music table, you call up the USR function. The computer will then start at the beginning of the table and play until it sees a duration of 0 (make sure the 0 is there!!! Otherwise, the computer will try to play whatever is in memory, after the music table—it may also never return to BASIC!).

When the computer encounters the duration 0, it will return to the next BASIC instruction following the USR command. The music can be played as often as you like by calling the USR function. The music table can also be changed by poking different values at any time during the BASIC program (just remember the computer starts at the beginning of the table every time, and it has to have the duration 0 to get back to BASIC).

NOTE: If you don't already have an amplifier connected to your computer, it's very simple to do. Just buy a small amplifier (RS 277-1008, about $12.00), and plug the cable from the computer (which normally goes to the aux jack on the cassette recorder) into the amplifier. Now you have sound!!!

Program 1. (16K non-disk)

```
1 DATA 62, Ø2, 211, 255, 221, 33, 188, 127, 221, 78,
     ØØ, 22, 1Ø, 121, 183, 2Ø2, 58, 127, 221, 126,
     Ø1, 183, 194, 22, 127, 221, 54, Ø1, 3Ø, 3Ø, Ø2,
     195, 24, 127, 3Ø, Ø1, 221, 7Ø, Ø1, 123, 211, 255
2 DATA 16, 254, 221, 7Ø, Ø1, 62, Ø2, 211, 255, 16,
     254, 21, 194, 24, 127, 22, 1Ø, 13, 194, 24, 127,
     221, 35, 221, 35, 195, 252, 126, 62, Ø2, 211,
     255, 2Ø1
3 FOR X=325ØØ TO 32574
     : READ Y
     : POKE X, Y
     : NEXT
     : POKE 16527, 126
     : POKE 16526, 244
4 REM  START MUSIC TABLE AT POKE 327ØØ:USE DURATION,
     TONE:RANGE 1-255; SHORT-LONG; HI-LO:USE DURATION
     Ø TO RETURN TO BASIC:USE TONE Ø FOR REST:MEM
     SIZE 32499
```

Program 2. (48K disk)

```
1 DATA 62, Ø2, 211, 255, 221, 33, 12Ø, 255, 221, 78,
     ØØ, 22, 1Ø, 121, 183, 2Ø2, 9Ø, 255, 221, 126,
     Ø1, 183, 194, 54, 255, 221, 54, Ø1, 3Ø, 3Ø, Ø2,
     195, 56, 255
2 DATA 3Ø, Ø1, 221, 7Ø, Ø1, 123, 211, 255, 16, 254,
     221, 7Ø, Ø1, 62, Ø2, 211, 255, 16, 254, 21, 194,
     56, 255, 22, 1Ø, 13, 194, 56, 255, 221, 35, 221,
     35, 195, 28, 255, 62, Ø2, 211, 255, 2Ø1
3 DEFUSR1=-236
     : FOR V=653ØØ TO 65374
     : READ V1
     : POKE (-1*(65536-V)), V1
     : NEXT
4 'START MUSIC TABLE AT POKE-136 TO Ø:USE DURATION,
     TONE:      RANGE 1-255 ;SHORT-LONG; HI-LO:USE
     DURATION Ø TO  RETURN
TO BASIC:USE TONE Ø FOR REST:USR1(X):DISABLE
     INTERRUPTS
FOR PURE TONE:MEM SIZE 65299
```

Addition Drill and Drawing Pad

Adam Roth
35-10 150 Street
Flushing, NY 11354

I have contributed two games for Models I and III, Addition Drill, a math teacher for addition facts and Drawing Pad, which uses the arrow keys to draw pictures.

Program 1. Addition Drill
This program will provide drill and practice for addition facts up to 100 + 100.

```
1ØØØØ CLS
1ØØØ1 PRINT
     : PRINT
1ØØØ2 INPUT "INPUT YOUR NAME PLEASE"; A$
     : PRINT "THANK YOU", A$
```

```
1ØØØ3 PRINT " THERE IS ONE TYPE OF FUNCTION
     POSSIBLE."
1ØØØ4 PRINT "            1) ADDITION"
1ØØØ5 PRINT
     : PRINT
1ØØØ6 INPUT "HIT ENTER TO BEGIN THE GAME OF
     MATHEMATICS"; A
1ØØØ7 A=RND(1ØØ)
     : B=RND(1ØØ)
     : C=A+B
1ØØØ8 CLS
1ØØØ9 PRINT
     : PRINT
     : PRINT
1ØØ1Ø PRINT "ONE HINT IS THAT IF YOU DO NOT KNOW THE
     ANSWER THEN HIT THE ENTER KEY"
1ØØ11 PRINT A; "PLUS"; B; "EQUALS"
     : INPUT D
1ØØ12 IF C=D THEN PRINT "YOU ARE RIGHT" ELSE PRINT
     "YOU ARE WRONG. THE ANSWER IS"; C
     : IF C<>D THEN Z=Z+1
     : GOSUB 1ØØ18
     : IF N>Ø THEN 1ØØØ7
1ØØ13 IF C<>D THEN Z=Z+1 ELSE X=X+1
     : GOSUB 1ØØ15
1ØØ14 GOTO 1ØØØ7
1ØØ15 PRINT @Ø, "WRONG ANSWERS"; Z
     : PRINT @45, "RIGHT ANSWERS"; X
1ØØ16 FOR I=Ø TO 1ØØØ
     : NEXT I
1ØØ17 RETURN
1ØØ18 PRINT @Ø, "WRONG ANSWERS"; Z
     : PRINT @45, "RIGHT ANSWERS"; X
1ØØ19 IF Z+X=1Ø THEN GOSUB 1ØØ22
1ØØ2Ø N=N+1:
1ØØ21 FOR I=Ø TO 1ØØØ
     : NEXT I
     : RETURN
1ØØ22 CLS
1ØØ23 B$=INKEY$
     : IF B$=""THEN 1ØØ23 ELSE 1ØØ24
1ØØ24 P=P+1
     : IF P=1Ø THEN PRINT
1ØØ25 X=Ø
     : Z=Ø
     : RETURN
```

Program 2. Drawing Pad

```
Ø CLS
     : CLEAR 2ØØØ
1 GOSUB 2ØØØ
2 FOR I=1 TO 1ØØØ
     : NEXT I
     : CLS
3 INPUT "DO YOU NEED INSTRUCTIONS (Y/N)"; B$
4 IF B$="N" THEN 5 ELSE 1ØØ
5 CLS
6 X=63
7 Y=25
8 SET (X, Y)
     : GOTO 1Ø
9 FOR X=Ø TO 127
     : FOR Y=Ø TO 47
     : IF POINT(X,Y)=1 THEN PRINT X, Y ELSE OUT 255,9
     : NEXT Y, X
1Ø A$=INKEY$
     : IF A$=CHR$(9) THEN X=X+1
     : SET (X, Y)
2Ø IF A$=CHR$(91) THEN Y=Y-1
     : SET (X, Y)
3Ø IF A$=CHR$(8)  THEN X=X-1
     : SET (X, Y)
4Ø IF A$=CHR$(1Ø) THEN Y=Y+1
     : SET (X, Y)
5Ø IF A$="6" THEN X=X+1
     : RESET(X, Y)
6Ø IF A$="8" THEN Y=Y-1
     : RESET(X, Y)
7Ø IF A$="4" THEN X=X-1
     : RESET(X, Y)
8Ø IF A$="2" THEN Y=Y+1
     : RESET(X, Y)
9Ø IF A$=CHR$(31) THEN GOTO 5
91 IF A$="S" THEN GOSUB 3ØØØ
92 IF A$="R" THEN 3ØØ4
95 GOTO 1Ø
1ØØ PRINT
11Ø PRINT A$
     : PRINT" BY USING THE APPROPRIATE ARROW KEYS YOU
     CAN DRAW"
     : PRINT A$
12Ø PRINT A$
     : PRINT " THE APPROPRIATE SHAPE OR OBJECT THAT
     YOU WANT TO CREATE!"
     : PRINT A$
13Ø PRINT A$
     : PRINT "BY PRESSING <S> YOU CAN SEARCH THE
     SCREEN FOR YOUR PICTURE AND"
     : PRINT A$
14Ø PRINT A$
     : PRINT "SAVE IT ON YOUR TAPE RECORDER. BY
     PRESSING <R> YOU CAN RECREATE"
     : PRINT A$
15Ø FOR I=Ø TO 5ØØØ
     : NEXT I
16Ø CLS
     : PRINT A$
     : PRINT "YOUR PICTURE. WHEN USING THE CORRECT
     ARROW KEYS YOU CAN ALSO"
     : PRINT A$
17Ø PRINT A$
     : PRINT "ERASE YOUR LINES BY USING        8
     4    6                        2"
     : PRINT A$
1998 FOR I=Ø TO 2ØØØ
     : NEXT I
1999 GOTO 5
2ØØØ A$=STRING$(63, 191)
     : PRINT @384, " "
     : PRINT A$
     : PRINT CHR$(191); "-----------C O S M I  C  S O
     F T W A R E---------"; CHR$(191)
2ØØ1 PRINT A$;
2ØØ2 FOR I=Ø TO 2ØØØ
     : NEXT I
     : CLS
2ØØ3 PRINT @384, " "
     : PRINT A$
     : PRINT CHR$(191); "----------------P   R  E  S
     E  N  T  S----------------"; CHR$(191)
     : PRINT A$;
     : FOR I=Ø TO 2ØØØ
     : NEXT I
     : CLS
2ØØ4 PRINT @384, "  "
     : PRINT A$
     : PRINT CHR$(191); "-----------------D R A W I N
     G  P A D---------------"; CHR$(191)
2ØØ5 : PRINT A$
     : FOR I=Ø TO 2ØØØ
     : NEXT I
     : CLS
     : RETURN
3ØØØ DIM D(16383-1536Ø)
     : Q$=CHR$(32)
     : DIM V(1Ø23)
3ØØ1 FOR V=1536Ø TO 16383
     : M=M+1
     : IF M=1Ø23 THEN 3ØØ2 ELSE P=Ø
     : D(M)= PEEK (V)
     : A= PEEK (15423)
     : PRINT @63, "*"
```

```
     : POKE 15423, A
     : NEXT V:
3002 FOR V=0 TO 1023
     : IF D(V)=32 THEN P=0 ELSE PRINT #-1, D(V), I
     : NEXT V
3003 RETURN
3004 CLS:
3005 FOR V=0 TO 1023
     : INPUT #-1, D(V), I
     : POKE I, D(V)
     : NEXT V
```

Word Counter

Joseph McCarthy
203 Lee Street
Brundidge, AL 36010

"Class, please make that essay five-hundred words.", the teacher said in a cold stern voice. Well, now that you have your assignment, you'll find this program a big help.

The Word Counter allows you to determine the number of words in a Model III SCRIPSIT file. The file may be saved in either the ASCII or non-ASCII format. I have found that the ASCII format gives a more accurate word count. Since this program is in BASIC, you will find that it runs rather slowly. I was able to process a 688 word document in 1 minute and 36 seconds.

After your file has been processed, a menu will appear. This menu will allow you to go back to BASIC, TRSDOS, The Word Counter, or SCRIPSIT. The return to SCRIPSIT from BASIC is accomplished by using CMD"I". The program will work as listed for the Model III, but Model I owners will have to make the following changes:

Delete lines 370, 420, and 430.

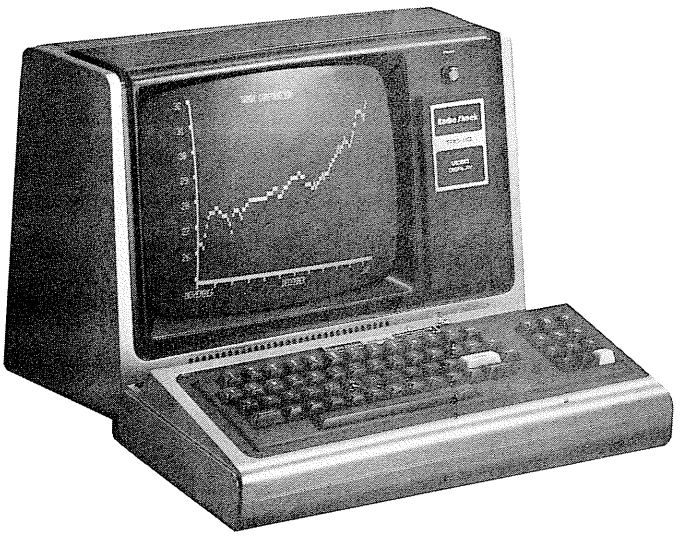
Change line 400 to: 400 POKE A, 94.

Change line 440 to: 440 IF B>2 THEN 450 ELSE IF PEEK(14400)=16 THEN B=B+1:A=A+64.

Change line 480 to: 480 ON B GOTO 10, 490, 500.

Oh, by the way, according to The Word Counter this article has 244 words.

```
10 CLS
     : CLEAR 1500
20 BP=1
30 PRINT TAB(24) "The Word Counter"
     : PRINT
40 INPUT "Which document should I process"; FILE$
50 OPEN "I", 1, FILE$
60 GOSUB 160
70 IF C$=" " THEN IN=0
     : GOTO 60
80 IF IN<>0 THEN GOTO 60
90 IN=1
     : CT=CT+1
100 GOTO 60
110 PRINT
     : A$="I found"+STR$(CT)+" words in "+FILE$+"."
     : PRINT TAB(32-LEN(A$)/2); A$
120 CLOSE
130 PRINT @ 976, "Please press <ENTER> to continue.";
140 IF PEEK(14400)<>1 THEN GOTO 140
150 GOTO 300
160 IF BP>LEN(B$) THEN 210
170 IF BP<LEN(B$) THEN 240
180 C$=" "
190 BP=BP+1
200 RETURN
210 BP=0
220 ON ERROR GOTO 110
230 INPUT #1, B$
240 BP=BP+1
250 C$=MID$(B$, BP, 1)
260 RETURN
300 CLS
310 PRINT TAB(24) "The Word Counter"
     : PRINT
     : PRINT TAB(24) "*** Main Menu ***"
320 PRINT STRING$(64, "=")
330 PRINT @ 640, STRING$(64, "=")
340 PRINT @ 337, " 1. Process another document"
350 PRINT @ 401, " 2.  Return to TRSDOS Ready"
360 PRINT @ 465, " 3.  Return to BASIC READY"
370 PRINT @ 529, " 4.   Return to SCRIPSIT"
380 PRINT @ 704, "Please select an option by using
    the <ARROW-KEYS>,"
     : PRINT "and then press <ENTER>."
390 A=15696
     : B=1
400 POKE A-2, 244
     : POKE A-1, 245
     : POKE A, 246
410 POKE A-64, 128
     : POKE A+64, 128
420 POKE A-65, 128
     : POKE A+63, 128
430 POKE A-66, 128
     : POKE A+62, 128
440 IF B>3 THEN 450 ELSE IF PEEK(14400)=16 THEN B=B+1
     : A=A+64
450 IF B<2 THEN 460 ELSE IF PEEK(14400)=8 THEN B=B-1
     : A=A-64
460 IF PEEK(14400)=1 THEN 480
470 GOTO 400
480 ON B GOTO 10, 490, 500, 510
490 CLS
     : CMD"S"
500 NEW
510 CLS
     : CMD"I", "SCRIPSIT/CMD"
```



PC-2 Assembly Language–Part 3

By Bruce Elliott

This is the third in a series of articles which will describe the MPU (microprocessor unit) used in the Radio Shack PC-2 pocket computer. It is our intention to include specific information about the 8-bit CMOS microprocessor, the machine code used by the microprocessor, as well as information about the PC-2 memory map, and certain ROM calls which are available. Please realize that much of what we are talking about refers to the overall capabilities of MPU, and does not imply that all of these things can be done with a PC-2.

The information provided in these articles is the only information which is available. We will try to clarify any ambiguities which occur in the articles, but can not reply to questions outside the scope of these articles. Further, published copies of TRS-80 Microcomputer News are the only source of this information, and we will not be maintaining back issues. Parts One and Two of this series were published in the March and April issues, respectively.

JUMPS/BRANCHES

BCH—Causes a relative jump to a new program area that is determined by adding/subtracting the immediate value i to/from the program counter P.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BCH+i	P+i→P	8E i	2	8
BCH–i	P–i→P	9E i	2	9

BCR—Conditional relative jump instruction. The relative jump is made when "C=0". If "C=1", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BCR+i	if C=0, P+i→P	81 i	2	8-11
BCR–i	if C=0, P–i→P	91 i	2	8-11

Comment—If C=1, no jump

BCS—Conditional relative jump instruction. When the condition "C=1" is met, a relative jump is made to the program area that is found after adding/subtracting the immediate value i to/from the program counter P. If "C=0", control proceeds to the next instruction without making the relative jump. It causes no flag change.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BCS+i	if C=1, P+i→P	83 i	2	8-11
BCS–i	if C=1, P–i→P	93 i	2	8-11

Comments—if C=0, no jump

BHR—A relative jump is made when "H=0". If "H=1", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BHR+i	if H=0, P+i→P	85 i	2	8-11
BHR–i	if H=0, P–i→P	95 i	2	8-11

Comment—if H=1, no jump

BHS—A relative jump is made when "H=1". If "H=0", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BHS+i	if H=1, P+i→P	87 i	2	8-11
BHS–i	if H=1, P–i→P	97 i	2	8-11

Comment—if H=0, no jump

BVR—A relative jump is made when "V=0". If "V=1", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BVR+i	if V=0, P+i→P	8D i	2	8-11
BVR–i	if V=0, P–i→P	9D i	2	8-11

Comment—if V=1, no jump

BVS—A relative jump is made when "V=1". If "V=0", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BVS+i	if V=1, P+i→P	8F i	2	8-11
BVS–i	if V=1, P–i→P	9F i	2	8-11

Comment—if V=0, no jump

BZR—A relative jump is made when "Z=0". If "Z=1", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BZR+i	if Z=0, P+i→P	89 i	2	8-11
BZR–i	if Z=0, P–i→P	99 i	2	8-11

Comment—if Z=1, no jump

BZS—A relative jump is made when "Z=1". If "Z=0", control proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
BZS+i	if Z=1, P+i→P	8B i	2	8-11
BZS–i	if Z=1, P–i→P	9B i	2	8-11

Comment—if Z=0, no jump

JMP—Causes a jump to a new program area implied by the immediate value in the second and third bytes. It causes no flag change.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
JMP i,j	i→PH, j→PL	BA i j	3	12

LOP—This instruction causes a relative jump to a new program area if, when UL is reduced by 1, no borrow occurs (i.e., UL remains positive or zero). The new program area is determined by subtracting the immediate value i from P. If a borrow occurs when UL is reduced by 1, no jump takes place and execution proceeds to the next instruction. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
LOP UL,i	UL – 1→UL	88 i	2	8-11

Comment—if borrow = 1, no jump; if borrow = 0, P – i→P

CALLS

SJP—Makes a subroutine jump to the address specified by the immediate values i and j. At the same time, the address of the next instruction is stored in the stack. It causes no flag changes.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
SJP	PL→(S), PH→(S – 1), S – 2→S, i→PH, j→PL	BE i j	3	19

VCR—Conditional vector subroutine jump. When "C = 0", the vector subroutine jump is performed. If "C = 1", the control proceeds to the next instruction. The Z flag is reset after the jump. VCR uses FF00 through FFF6 as its vector address table, and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VCR i	if C = 0, PH→(S – 1), PL→(S) (FFab)→PH, (FFab + 1)→PL S – 2→S	C1 i	2	8-21

Comment—if C = 1, no jump, ab = Hex digits in i

VCS—Conditional vector subroutine jump. When "C = 1", it performs the vector subroutine jump. If "C = 0", the control proceeds to the next instruction. The Z flag is reset after the jump. VCS uses FF00 through FFF6 as its vector address table and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VCS i	if C = 1, PH→(S – 1), PL→(S) (FFab)→PH, (FFab + 1)→PL S – 2→S	C3 i	2	8-21

Comment—if C = 0, no jump, ab = Hex digits in i

VEJ—Vector subroutine jump. VEJ is a one byte instruction which makes a subroutine jump based on a vectored address. The vector table is located in memory from FF00 to FFF6. The Z flag is reset after the vector jump is executed.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VEJ (ab)	PL→(S), S – 1→S			
VEJ (C0)	PH→(S), S – 1→S	C0	1	17
VEJ (C2)	(FFab)→PH	C2	1	17
VEJ (C4)	(FFab + 1)→PL	C4	1	17
VEJ (C6)		C6	1	17
VEJ (C8)		C8	1	17
VEJ (CA)		CA	1	17
VEJ (CC)		CC	1	17
VEJ (CE)		CE	1	17
VEJ (D0)		D0	1	17
VEJ (D2)		D2	1	17
VEJ (D4)		D4	1	17
VEJ (D6)		D6	1	17
VEJ (D8)		D8	1	17
VEJ (DA)		DA	1	17
VEJ (DC)		DC	1	17
VEJ (DE)		DE	1	17
VEJ (E0)		E0	1	17
VEJ (E2)		E2	1	17
VEJ (E4)		E4	1	17
VEJ (E6)		E6	1	17
VEJ (E8)		E8	1	17
VEJ (EA)		EA	1	17
VEJ (EC)		EC	1	17
VEJ (EE)		EE	1	17
VEJ (F0)		F0	1	17
VEJ (F2)		F2	1	17
VEJ (F4)		F4	1	17
VEJ (F6)		F6	1	17

Comment—Where, "ab" is the instruction code of VEJ.

VHR—Conditional vector subroutine jump. When "H = 0", the vector subroutine jump is performed. If "H = 1", the control proceeds to the next instruction. The Z flag is reset after the jump. VHR uses FF00 through FFF6 as its vector address table and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VHR i	if H = 0, PH→(S – 1), PL→(S) (FFab)→PH, (FFab + 1)→PL S – 2→S	C5 i	2	8-21

Comment—if H = 1, no jump, ab = Hex digits in i

VHS—Conditional vector subroutine jump. When "H = 1", it performs the vector subroutine jump. If "H = 0", the control proceeds to the next instruction. The Z flag is reset after the jump. VHS uses FF00 through FFF6 as its vector address table and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VHS i	if H = 1, PH→(S – 1), PL→(S) (FFab)→PH, (FFab + 1)→PL S – 2→S	C7 i	2	8-21

Comment—if H = 0, no jump; ab = Hex digits in i

VMJ—Vector subroutine jump. VMJ is the subroutine jump that branches to a vectored address, of which the high order byte is composed of "FF", and low order byte is composed of the immediate value i. Note that the Z flag is reset after the vector jump, when VMJ is executed. VMJ uses FF00 through FFF6 as its vector address table, and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VMJ i	PL→(S), S – 1→S PH→(S), S – 1→S (FFab)→PH (FFab + 1)→PL	CD i	2	20

Comments—ab = Hex digits in i

VVS—Conditional vector subroutine jump. When "V = 1", it performs the vector subroutine jump. If "V = 0", the control proceeds to the next instruction. The Z flag is reset after the jump. VVS uses FF00 through FFF6 as

its vector address table and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VVS i	if V=1, PH→(S−1), PL→(S) (FFab)→PH, (FFab+1)→PL S−2→S	CF i	2	8-21

Comment—if V=0, no jump; ab = Hex digits in i

VZR—Conditional vector subroutine jump. When "Z=0", the vector subroutine jump is performed. If "Z=1", the control proceeds to the next instruction. The Z flag is reset after the jump. VZR uses FF00 through FFF6 as its vector address table and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VZR i	if Z=0, PH→(S−1), PL→(S) (FFab)→PH, (FFab+1)→PL S−2→S	C9 i	2	8-21

Comment—if Z=1, no jump; ab = Hex digits in i

VZS—Conditional vector subroutine jump. When "Z=1", it performs the vector subroutine jump. If "Z=0", the control proceeds to the next instruction. The Z flag is reset after the jump. VZS uses FF00 through FFF6 as its vector address table, and the values 00 through F6 are valid for the immediate value.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
VZS i	if Z=1, PH→(S−1), PL→(S) (FFab)→PH, (FFab+1)→PL S−2→S	CB i	2	8-21

Comment—if Z=0, no jump; ab = Hex digits in i

RETURNS

RTI—Return instruction from the interrupt subroutine to the main routine. All flags are subject to change.

Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
RTI	(S+1)→PH, (S+2)→PL, (S+3)→T, S+3→S	8A	1	14

RTN—Return instruction from a subroutine to the calling routine. RTN causes no changes in the flags.

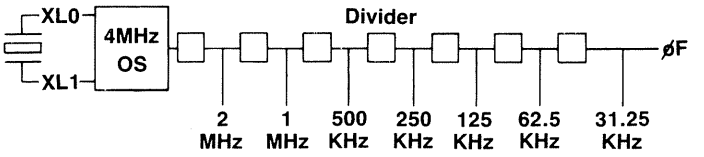
Mnemonic	Symbolic Operation	Hex Op-Code	Byte	Cycle
RTN	(S+1)→PH, (S+2)→PL, S+2→S	9A	1	11

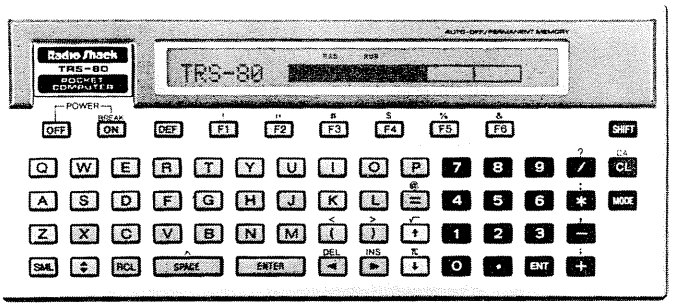
TIMER

The timer is composed of a 9-bit polynomial counter and the time duration can be set using the AM0 and AM1 instructions. This counter is in operation at all times, so it needs to be set to 000 (Hex) before being used. A timer interrupt request can be generated when the content of the counter is 1FF (Hex), if Interrupt Enable IE is on.

When a timer interrupt occurs, interrupt processing begins at the address specified in addresses FFFA and FFFB.

When a 4MHz crystal oscillator is used, the clock produces a oF of 31.25KHz with a cycle of 32 microseconds. In other words, the timer counter is incremented once every 32 microseconds.





National Computer Camp

National Computer Camp (NCC), which is believed to be the first computer camp in this country, is powering up for another series of computer camps for kids from ages nine to eighteen.

Campers at National Computer Camps learn to program the computer through a hands-on approach along with ample time on, and access to, the computer—two to three campers per computer. In the process, the campers come to understand the potential as well as the limitations involved in using computers.

1983 will be the sixth season of NCC, and for the first time, the camp will be held in three separate locations: Simsbury, CT; Atlanta, GA; and St. Louis, MO.

The sessions will be:

July 3—July 8
July 10—July 15
July 17—July 22
July 24—July 29
July 31—August 5

National Computer Camps
P.O. Box 585
Orange, CT 06477
1-203-795-9667

FLASH—Fourth Location Now Available

Dr. Zabinski has just informed us of a fourth National Computer Camp location for this summer. The fourth camp will be at Linfield College in McMinnville, Oregon. For further information, contact NCC at the address and phone number shown above.

PC-2 INPUT Syntaxes

By Peter Levy

One of the most powerful (and probably least understood) features of the PC-2 is its handling of BASIC's INPUT statement. It is one of the most adaptable and powerful implementations of INPUT ever published in any version of BASIC. Of course, the more a programming statement can do, the more there is to learn about it, and the more ways there are to get messed up using it.

FUNDAMENTAL INPUT SYNTAX

At its simplest level, the PC-2's syntax for the INPUT statement is quite conventional: 'INPUT X' displays a "?" (to show the PC-2 is asking for information) and loads a value from keyboard input into the specified variable (in this case, X). Simple and conventional.

INPUT WITH PROMPT

Like most other BASICs, the PC-2 can display a prompt from within an INPUT statement. Unlike most, there are two separate ways to do this.

The first way displays a prompt and a cursor (again, this is to show that the PC-2 is asking for something—to be technical, the cursor indicates that the keyboard is "active"). The syntax for this form of INPUT is 'INPUT "prompt";X'. When this is executed, "prompt" is displayed, and the user's input is displayed as he types it (it is "echoed").

The second way is pretty similar, the differences being that no cursor is displayed with the prompt, and, (more noticeable), the display is cleared the instant the first key is pressed in reply to the prompt. The syntax for this form of INPUT is 'INPUT "prompt",X'. The only difference is that a comma, rather than a semicolon, separates the prompt and variable.

INPUT OF SEVERAL VARIABLES

The PC-2 offers another variant of the INPUT statement which allows more than one variable to be loaded with a single INPUT command. The general syntax used is INPUT A,B,C, ... Although this syntax is conventional, its execution is not. The user must press ⟨**ENTER**⟩ after EACH value he types. This is different from Radio Shack's desktop models, for which all the INPUT values are typed on one line and then ⟨**ENTER**⟩ is only pressed once, at the end.

Each time the PC-2 asks for a next INPUT value, the "?" prompt is displayed again to show that the keyboard is still active. In fact INPUT A,B,C is exactly the same as three separate single-variable INPUTs: INPUT A, INPUT B, and INPUT C.

SUMMARY OF INPUT RULES

Now let's set forth a few nice simple general rules about all this.

First, any INPUT is started with the word INPUT.

Second, consecutive variables to be loaded are listed after INPUT and are separated by commas. This is called an input list.

Third, any input variable within the input list may be preceded with a quoted text which will be displayed as a prompt when the PC-2 requests that particular variable. The prompt and the variable name should be separated by either a comma or a semicolon. If a comma is used, the prompt disappears when you start entering data; if a semicolon is used, the prompt remains displayed as you type.

Fourth, as long as you follow these three general rules, you can get away with almost any combination of the various types of INPUTs you can think of.

These examples are all valid uses of INPUT:

```
INPUT X
     :  INPUT Y
INPUT "ENTER X ";X
     :  INPUT "ENTER Y ";Y
INPUT "ENTER X, THEN Y ";X,Y
INPUT "ENTER X ";X,"ENTER Y ";Y
INPUT "ENTER X, THEN Y ";X,Y,"NOW ENTER Z ";Z
```

EVALUATION OF NUMERIC INPUT

One of the most powerful features of the PC-2's INPUT statement is its ability to evaluate expressions. This means that if the PC-2 tries to INPUT a numeric variable and the user enters something besides a nice simple number, the PC-2 will try to calculate a numeric value for whatever was entered, instead of declaring an error.

For example, if SQR(Y) was entered as numeric input in response to 'INPUT "ENTER X"; X' for most computers they would think something like "Hey, that stuff isn't numbers!" and would declare an error. The PC-2 will evaluate that input as a BASIC expression: the square root of Y. It will call up whatever Y is equal to, take the square root, and assign that computed value to the input variable.

This is REALLY powerful. Among other things, you can enter oddball numbers (like SQR(2) or 1/3) very accurately and easily, instead of having to remember and then type 1.414213562 and .333333333.

The disadvantage of all this is that the PC-2 can't detect "garbage" input. If you enter "GROWRP" as a numeric value for most computers, they'll spit it out; but the PC-2 will try to figure out what it all means. (In this case, it would assign a numeric value from the variable GR.)

Incidentally, if what you enter is REALLY garbage, the PC-2 will tell you: ERROR 1 will be displayed. Press ⟨**CLEAR**⟩ to erase the error message and try again.

IRA Account

Harold R. Wright
2900 Powerline Road, Lot 39
Haines City, FL 33844

Here is a simple program for the pocket computer. The program is used for estimating the value of an I.R.A. account with equal yearly deposits. Press (SHFT) and (C). Then answer the questions on the display. Interest may be compounded for 1 to 365 times per year. The print out shows the value for each year up to the number of years requested.

I.R.A. ACCOUNT
$2000. DEPOSIT
12% INTEREST
COMPOUNDED 4.
TIMES PER YEAR
FOR 5. YEARS

YEAR	BALANCE
1.	$2251.02
2.	$4784.56
3.	$7636.08
4.	$10845.49
5.	$14457.71

```
1Ø "C"
     : CLEAR
     : PRINT "I.R.A. ACCOUNT"
     : PRINT "  "
2Ø INPUT "DEPOSIT/YEAR $"; X, "% INTEREST?"; Y, "#
   COMPOUNDED / YEAR"; Z
3Ø INPUT "NO.OF YEARS?"; V
4Ø PRINT "$"; X; " DEPOSIT "
     : PRINT Y; "%INTEREST"
5Ø PRINT "COMPOUNDED "; Z
     : PRINT "TIMES PER YEAR"
     : PRINT "FOR "; V; " YEARS"
6Ø PRINT "    "
7Ø PRINT "YEAR BALANCE"
8Ø PRINT "--------------"
9Ø FOR U=1 TO V
1ØØ W=(1+Y/1ØØZ)^Z*(W+X)
     : T=INT (1ØØW+.5)/1ØØ
11Ø PRINT U; "   $"; T
     : NEXT U
12Ø PRINT "  "
     : PRINT "  "
     : PRINT "  "
13Ø END
```

GMT Converter

Charles J. Blatt
916 Melrose Street
New Lenox, IL 60451

Here is a PC-2 program that will convert EST, CEN, MNT, or PAC to Greenwich Mean Time or it will convert Greenwich Mean Time to EST, CEN, MNT or PAC.

```
1Ø PAUSE "<_24-HR_GMT_CONVERTER_>"
     : CLEAR
2Ø INPUT "WANT TO OR FROM GMT?_";Z$
3Ø INPUT "TIME BEING CONVERTED: _";B
4Ø PAUSE "YOUR TIME ZONE"
5Ø INPUT "IS: EST-CEN-MNT-PAC: _";A$
6Ø IF A$="EST" LET C=5
7Ø IF A$="CEN" LET C=6
8Ø IF A$="MNT" LET C=7
9Ø IF A$="PAC" LET C=8
1ØØ IF Z$=" FROM " T.15Ø
11Ø D=B+C
     : PAUSE "THE TIME IS: "
     : BEEP 2
12Ø IF D>24 LET D=D-24
13Ø PRINT USING "###.##";    D;"_ _ GMT"
14Ø GOTO 2Ø
15Ø D=B+(24-C)
     : PAUSE "THE TIME IS: "
     : BEEP 2
16Ø IF D>24 LET D=D-24
17Ø PRINT USING "###.##"; D ;"_ _";A$
18Ø GOTO 2Ø
```

Enter time as: 12.45 or 1.25, etc.

X/Y Graph

David Baker
3280 Turgot Circle
Evendale, OH 45241

Rectangular Coordinate System Graphics Subroutine:

A most common need of business, science, and engineering problem solvers is to be able to display computation results on a rectangular coordinate system. With the TRS-80 Pocket Computer PC-2, and printer interface, such a task can be nicely accomplished for report presentation within the framework of an eight and one-half inch boundary.

The PC-2 BASIC subroutine listing (figure 2) performs the task of graphics display of a rectangular ordinate-abscissa system with the following features:

1. Ordinate and abscissa numerical graduations are automatically scaled to accommodate the largest 'X' and 'Y' value outputs to be displayed.
2. Graduation and scaling presentations are conditioned for optimum display size, with room to spare for aside data statements.

Prior to subroutine execution, the following variable must be defined by the mainline program:

1. YM; (Y-axis maximum value). Maximum computation value to be displayed, 'Y' ordinate.
2. XM; (X-axis maximum value). Maximum computation value to be displayed, 'X' abscissa.

In addition to program execution, a call-up of subroutine variables 'YD' and 'XD' will reveal axis numerics, denominator scaling factors. Whole value factors were introduced as a means to optimize overall presentation of axis numbers.

YD = 'Y' ordinate axis self scale divisor value
XD = 'X' abscissa axis self scale divisor value

Multiplication of 'X' or 'Y' axis values displayed by 1/YD or 1/XD, where appropriate, will give rise to actual axis values.

Coordinate cursor locating within the envelope of the displayed rectangular system (cartesian system) can be simply accomplished by the execution of the line function:

GLCURSOR (Y*198/O1,-X*500/A1)
where: 'Y' = Y axis coordinate position
'X' = X axis coordinate position

NOTE: This program uses three character variable names. The PC-2 only recognizes the first two characters. To use this routine, change 830 to RETURN.

```
5ØØ X1=YM, YD=1, Y2=YM
5Ø1 IF Y2<99999 GOTO 5Ø5
```

```
502 YD=YD/10, Y2=Y2/10, X1=X1/10
503 GOTO 501
505 GOTO 585
510 GRAPH
515 GLCURSOR (0, 0)
     : COLOR 0
520 GLCURSOR (18, 0)
     : SORGN
525 LINE -(198, 0)
530 GLCURSOR (0, 0)
535 FOR XA1=0 TO 190 STEP 10
540 GLCURSOR (XA1, 0)
545 LINE -(XA1, 5)
550 GLCURSOR (XA1, 40)
555 CSIZE 1
560 ROTATE 1
565 O1=XA1*SF1
570 LPRINT USING "######"; O1
575 NEXT XA1
580 GOTO 670
585 IF X1=0 THEN 660
590 IF X1>1.9 THEN 606
595 X1=X1*100
600 YD=YD*100
605 GOTO 590
606 IF X1>10 GOTO 610
607 X1=X1*10
608 YD=YD*10
610 XL1=190, SF1=.1
615 IF X1>SF1*XL1 GOTO 625
620 GOTO 635
625 SF1=SF1*10
630 GOTO 615
635 IF SF1=.1 GOTO 510
640 SF1=SF1/10
645 SF1=SF1+.1
650 IF X1<=SF1*XL1 GOTO 510
655 GOTO 645
660 PRINT "ILLEGAL ORDINATE MAXIMUM"
670 Y1=XM, XD=1, X2=XM
671 IF X2<999 GOTO 675
672 XD=XD/10, Y1=Y1/10, X2=X2/10
673 GOTO 671
675 GOTO 740
680 GLCURSOR (0, 0)
685 ROTATE 0
690 LINE -(0, -500)
695 GLCURSOR (0, 0)
700 FOR YA1=0 TO 500 STEP 10
705 GLCURSOR (0, -YA1)
710 LINE -(1, -YA1)
715 GLCURSOR (-24, -YA1)
720 A1=YA1*SF2
725 LPRINT USING "####"; A1
730 NEXT YA1
735 GOTO 825
740 IF Y1=0 THEN 815
745 IF Y1>5 THEN 761
750 Y1=Y1*100
755 XD=XD*100
760 GOTO 745
761 IF Y1>25 GOTO 765
762 Y1=Y1*10
763 XD=XD*10
765 YL1=500, SF2=.1
770 IF Y1>SF2*YL1 GOTO 780
775 GOTO 790
780 SF2=SF2*10
785 GOTO 770
790 IF SF2=.1 GOTO 680
795 SF2=SF2/10
800 SF2=SF2+.1
805 IF Y1<SF2*YL1 GOTO 680
810 GOTO 800
815 PRINT "ILLEGAL ABCISSA MAXIMUM"
825 GLCURSOR (0, 0)
830 STOP
```

Resistors

Charles M. MacLeod
10 North Crane Avenue
Taunton, MA 02780

Here is a program for the Pocket Computer (PC-2) that I thought your readers might enjoy.

```
1 GOSUB 3000
10 GRAPH
20 GLCURSOR (0, -10)
     : SORGN
30 LINE (0, 0)-(225, 0), 3, 2
40 GLCURSOR (0, 0)
     : TEXT
     : CSIZE 4
     : LF 1
45 COLOR 1
50 LPRINT "Parallel"
60 LPRINT "Resistors"
70 GRAPH
     : GLCURSOR (0, 20)
     : SORGN
80 LINE (0, 0)-(225, 0), 3, 2
90 GLCURSOR (0, -100)
     : SORGN
100 LINE (0, 0)-(50, 0)-(50, 25)-(150, 25), 0, 0
1000 REM
1010 FOR X=75 TO 130 STEP 5
1020 GLCURSOR (X, 30)
1040 LINE (X, 35)-(X, 15)
1050 NEXT X
1060 GLCURSOR (50, 0)
1070 LINE (50, 0)-(50, -25)-(150, -25)-(150, 25)
1080 GLCURSOR (150, 0)
1090 LINE (150, 00)-(200, 0)
2000 FOR X=75 TO 130 STEP 5
2010 GLCURSOR (X, -30)
2030 LINE (X, -35)-(X, -15)
2040 LINE (X, -35)-(X, -15)
2050 NEXT X
2060 GLCURSOR (55, 45)
     : CSIZE 2
     : COLOR 3
     : LPRINT USING #####.##"; R1; USING ; " ohms"
2070 GLCURSOR (55, -55)
     : CSIZE 2
     : COLOR 3
     : LPRINT USING "####.##"; R2; USING ; " ohms"
2080 GLCURSOR (0, -175)
     : SORGN
     : COLOR 1
2085 TEXT
     : USING "###.##"
2090 LPRINT "R1="; R1; "ohms"
     : LF 1
2100 LPRINT "R2="; R2; "ohms"
     : LF 1
2110 LPRINT "TR="; TR; "ohms"
2120 LF 2
2130 GRAPH
2140 LINE (0, 0)-(225, 0), 3, 2
2150 TEXT
     : CLEAR
     : GOTO 1
3000 INPUT "R1="; R1, "R2="; R2
3010 TR=1/(((R1+R2)/(R1*R2)))
3012 WAIT 0
     : USING "####.##"
3015 PRINT "TOTAL RESISTENCE="; TR
3020 RETURN
```

OKZAP

Dennis Tanner

This program is designed to let you edit material on your Color Computer Disks. Be aware that the process of editing material could render that material (or the entire diskette) unusable. If you don't know what you are editing, or what the results should be, DON'T EDIT IT!

Enter values for the Drive, Track, and Sector which you wish to see and/or modify.

The contents of the sector are displayed in Hex on the left side of the screen and in ASCII on the right side of the screen.

Use the (+) and (-) keys to page forward and back on the disk.

Press (J) to select another Drive, Track, and Sector.

Each sector is displayed in two parts. That is it takes two screens to show each sector. The right side of the screen gives you information on the current Drive, Track, Sector, and Part.

When the contents of the sector you wish to modify are shown, type (M) to get into the modify mode.

In the modify mode:

Arrow keys move "current location" number around on the screen.

To modify the current location, type the new hex value.

To work with ASCII values instead of Hex, type (SHIFT)(→). You can now modify the sector by typing ASCII values. If you wish to return to entering Hex values, use (SHIFT)(←).

Press (ENTER) to save the changes to disk.

Press (BREAK) to exit.

```
5 'WRITTEN BY DENNIS TANNER
10 'OKZAP 05/12/82
11 PCLEAR 1
     : CLEAR 1000
     : GOTO 100
15 CLS
     : PRINT @ 14, "OKZAP"
     : RETURN
20 DSKI$ D, T, S, A$, B$
     : RETURN
100 GOSUB 15
     : PRINT
     : INPUT "DRIVE"; D
     : INPUT "TRACK"; T
     : INPUT "SECTOR"; S
110 GOSUB 20
200 CLS
     : GOSUB 600
     : PRINT @ 447, "1";
     : FOR J=0 TO 15
     : FOR K=1 TO 8
     : A=ASC(MID$(A$, J*8+K, 1))
     : B=INT(A/16)
     : C=A-B*16
     : IF B>9 THEN B=B+7
210 IF C>9 THEN C=C+7
215 H$=CHR$(B+48)+CHR$(C+48)
217 IF A>31 THEN PRINT @ J*32+K+21, CHR$(A);  ELSE
     PRINT @ J*32+K+21, ".";
220 PRINT @ J*32+K*3-2-INT(K/2), H$;
     : NEXT K, J
300 I$=INKEY$
     : IF I$="" THEN 300
310 IF I$="; " THEN GOTO 400 ELSE IF I$="-" THEN 320
     ELSE IF I$="J" THEN 100 ELSE IF I$="M" THEN FL=0
     : GOSUB 800
     : GOTO 300 ELSE 300
320 S=S-1
     : IF S=0 THEN S=18
     : T=T-1
     : IF T=-1 THEN T=0
     : GOTO 300
330 GOSUB 20
400 CLS
     : GOSUB 600
     : PRINT @ 447, "2";
     : FOR J=0 TO 15
     : FOR K=1 TO 8
     : A=ASC(MID$(B$, J*8+K, 1))
     : B=INT(A/16)
     : C=A-B*16
     : IF B>9 THEN B=B+7
410 IF C>9 THEN C=C+7
415 H$=CHR$(B+48)+CHR$(C+48)
417 IF A>31 THEN PRINT @ J*32+K+21, CHR$(A);  ELSE
     PRINT @ J*32+K+21, ".";
420 PRINT @ J*32+K*3-2-INT(K/2), H$;
     : NEXT K, J
500 I$=INKEY$
     : IF I$="" THEN 500
510 IF I$="; " THEN 550 ELSE IF I$="M" THEN FL=1
     : GOSUB 800
     : GOTO 500
515 IF I$="J" THEN 100
520 IF I$="-" THEN 200 ELSE 500
550 S=S+1
     : IF S=19 THEN S=1
     : T=T+1
560 GOTO 110
600 PRINT @ 31, "D";
     : PRINT @ 159, "T";
     : PRINT @ 287, "S";
     : PRINT @ 62, D;
     : T$=RIGHT$(STR$(T), LEN(STR$(T))-1)
     : FOR J=1 TO LEN(T$)
     : PRINT @ J*32+159, MID$(T$, J, 1)
     : NEXT J
     : S$=RIGHT$(STR$(S), LEN(STR$(S))-1)
     : FOR J=1 TO LEN(S$)
     : PRINT @ J*32+287, MID$(S$, J, 1);
     : NEXT J
605 PRINT @ 415, "P";
610 RETURN
800 I$=INKEY$
     : V=0
     : J=0
     : K=1
805 LO=V*32+J*5
     : PRINT @ LO, RIGHT$(STR$(K), 1);
810 I$=INKEY$
     : IF I$="" THEN 810 ELSE IF I$=CHR$(13) THEN
     PRINT @ LO, " ";
     : DSKO$ D, T, S, A$, B$
     : RETURN
815 IF I$=CHR$(93) THEN 900
820 I=ASC(I$)
     : IF I=9 AND K=1 THEN K=3
     : GOTO 805 ELSE IF I=9 AND K=3 AND J<3 THEN
     PRINT @ LO, " ";
     : K=1
     : J=J+1
     : GOTO 805 ELSE IF I=9 AND K=3 AND J=3 AND V<15
     THEN PRINT @ LO, " ";
     : V=V+1
     : J=0
     : K=1
     : GOTO 805
830 IF I=8 AND K=3 THEN  K=1
     : GOTO 805 ELSE IF I=8 AND K=1 AND J>0 THEN
     PRINT @ LO, " ";
     : J=J-1
     : K=3
     : GOTO 805 ELSE IF I=8 AND K=1 AND J=0 AND V>0
```

```
     THEN PRINT @ LO, " ";
     : J=3
     : V=V-1
     : K=3
     : GOTO 8Ø5
84Ø IF I=1Ø AND V<15 THEN PRINT @ LO, " ";
     : V=V+1
     : GOTO 8Ø5 ELSE IF I=94 AND V>Ø THEN PRINT @ LO,
     " ";
     : V=V-1
     : GOTO 8Ø5
85Ø IF NOT(I>47 AND I<58) AND NOT(I>64 AND I<71) THEN
     8Ø5 ELSE IF I>31 THEN PRINT @ LO+K, I$;  ELSE
     STOP
     : PRINT @ LO+K, ".";
855 P1$=I$
     : K=K+1
     : PRINT @ LO, RIGHT$(STR$(K), 1);
86Ø I$=INKEY$
     : IF I$="" THEN 86Ø ELSE I=ASC(I$)
     : IF I<48 OR (I>57 AND I<65) OR I>7Ø THEN 86Ø
87Ø PRINT @ LO+K, I$;
     : P1$=P1$+I$
     : P1=VAL("&H"+P1$)
     : IF FL=Ø THEN MID$(A$, V*8+J*2+K/2)=CHR$(P1)
     ELSE MID$(B$, V*8+J*2+K/2)=CHR$(P1)
88Ø IF P1>31 THEN PRINT @ V*32+J*2+K/2+21, CHR$(P1);
     ELSE PRINT @ V*32+J*2+K/2+21, ".";
89Ø IF K=2 THEN K=3
     : GOTO 8Ø5 ELSE IF K=4 AND J<3 THEN PRINT @ LO,
     " ";
     : J=J+1
     : K=1
     : GOTO 8Ø5 ELSE IF K=4 AND J=3 AND V<15 THEN
     PRINT @ LO, " ";
     : V=V+1
     : J=Ø
     : K=1
     : GOTO 8Ø5 ELSE IF K=4  AND J=3 AND V=15 THEN
     PRINT @ LO, " ";
     : V=Ø
     : J=Ø
     : K=1
     : GOTO 8Ø5
895 STOP
9ØØ PRINT @ LO, " ";
     : L=INT(J*2+K/2)+1
9Ø5 PRINT @ V*32+21, RIGHT$(STR$(L), 1);
91Ø I$=INKEY$
     : IF I$="" THEN 91Ø ELSE I=ASC(I$)
     : IF I=9 AND L<8 THEN L=L+1
     : GOTO 9Ø5 ELSE IF I=9 AND L=8 AND V<15 THEN
     PRINT @ V*32+21, " ";
     : V=V+1
     : L=1
     : GOTO 9Ø5
915 IF I=13 THEN PRINT @ V*32+21, " ";
     : DSKO$ D, T, S, A$, B$
     : RETURN
92Ø IF I=8 AND L>1 THEN L=L-1
     : GOTO 9Ø5 ELSE IF I=8 AND L=1 AND V>Ø THEN
     PRINT @ V*32+21, " ";
     : V=V-1
     : L=8
     : GOTO 9Ø5
93Ø IF I=94 AND V>Ø THEN PRINT @ V*32+21, " ";
     : V=V-1
     : GOTO 9Ø5 ELSE IF I=1Ø AND V<15 THEN PRINT @
     V*32+21, " ";
     : V=V+1
     : GOTO 9Ø5
932 IF I=21 THEN PRINT @ V*32+21, " ";
     : J=INT((L-1)/2)
     : K=(L-J*2)*2-1
     : GOTO 8Ø5
94Ø IF I<32 OR I>9Ø THEN 9Ø5 ELSE PRINT @ V*32+21+L,
     I$;
     : IF  FL=Ø THEN MID$(A$, V*8+L, 1)=I$  ELSE IF
     FL=1 THEN MID$(B$, V*8+L, 1)=I$
942 X=INT(I/16)
     : Y=I-X*16
     : IF X>9 THEN X=X+7
944 IF Y>9 THEN Y=Y+7
945 Q$=CHR$(X+48)+CHR$(Y+48)
     : PRINT @ V*32+(L-1)*3+1-INT(L/2), Q$;
95Ø IF L<8 THEN L=L+1 ELSE IF L=8 AND DV<15 THEN
     PRINT @ V*32+21," ";
     : L=1
     : V=V+1
99Ø GOTO 9Ø5
1ØØØØ GOTO 1ØØØØ
2ØØØØ I$=INKEY$
     : IF I$ = "" THEN 2ØØØØ ELSE PRINT TI$;ASC(I$)
     : GOTO 2ØØØØ
3ØØØØ INPUT L
     : PRINT (L-1)*3+1-INT(L/2)
     : GOTO 3ØØØØ
```

Disk Editor Disassembler For Disk Color Computer

Ashok Basargekar
1423 North Cleveland Street
Orange, CA 92667

Thanks to Tomas Rokicki, for helping me to learn the 6809 Assembly Language, with the use of his 6809 Disassembler program in the Product Line Manager's News of November 1981 Issue of the Microcomputer News. I have used his program to study several simple Assembly Language programs written for the Color Computer. Many a time, while using his program I came across an "OM ERROR" message on the screen. But not any more. When I bought a Disk Drive for my Color Computer, I created a separate Direct Access Disk File for all the Data in his Program. I gave this file the name "SUFFIX.DAT". To create this file on the Disk, I used the listing no.2 given below.

I deleted all the DATA statements this way from his program. Next thing I had to do was, simply access my Disk File for the 6809 code, by refining his Disassembler program. I changed few lines here and there, and finally came up with following listing no.1 for my "EDIT ASM.BAS" program. The program can be used for disassembling as well as editing any machine code. For the prompt "ENTRY ADDRESS" enter the decimal address or Hex address with a pre-fix "H". In the EDIT mode the code appears with a Left Arrow in front of the line. If you want to change the code, type the new machine code and (ENTER). The typed machine code with its 6809 equivalent will appear on the next line. If it is correct, press either the UP or the DOWN ARROW. Otherwise you may enter any other desired machine code. By pressing (@) you can go backward or forward to any other address. By pressing the key (Q), you can exit from the EDIT or the DISASSEMBLER mode. I am using this method to write my Assembly Language sub-routines very successfully.

```
1Ø CLS
     : PRINT @33, "1:68Ø9 EDITOR"
     : PRINT @65, "2:68Ø9 DISASSEMBLER"
     : PRINT
     : PRINT "SELECT 1 OR 2"
     : INPUT QQ
     : IF QQ=2 THEN INPUT "1:SCREEN  2:PRINTER"; ZZ
```

```
2Ø F$="CODE.DAT"
     : G$="SUFFIX.DAT"
     : OPEN "D", #1, F$, 18
     : FIELD #1, 18 AS N$
     : OPEN "D", #2, G$, 6
     : FIELD #2, 6 AS I$
3Ø INPUT "ENTRY ADDRESS (DEFAULT IS 4Ø999) "; A$
     : IF A$="" THEN PI=4Ø999 ELSE IF MID$(A$, 1,
     1)="H" THEN PI=VAL("&"+A$) ELSE PI=VAL(A$)
4Ø BY=PEEK(PI)+1
5Ø IF BY<>17 AND BY<>18 THEN TE=Ø
     : GET #1, BY
     : BY$=N$
     : TY=VAL(BY$)
     : BY$=MID$(BY$, 2)
     : IF LEFT$(BY$, 1)=" " THEN 11Ø ELSE 7Ø
51 TE=1
     : BY=PEEK(PI+1)+1
     : GET #1, BY
     : BY$=N$
     : X=INSTR(BY$, "/")
     : IF X=Ø THEN 11Ø ELSE BY$=MID$(BY$, X+1)
     : IF PEEK(PI)=18 THEN X=INSTR(X, BY$, "/")
     : IF X=Ø THEN 11Ø ELSE 2ØØ
6Ø TY=VAL(BY$)
     : BY$=MID$(BY$, 2)
7Ø ON TY+1 GOSUB 35Ø, 18Ø, 21Ø, 22Ø, 29Ø, 3ØØ, 31Ø,
     32Ø, 33Ø, 34Ø
     : GOTO 12Ø
8Ø IF B$<>"C" THEN B$=INKEY$
     : IF B$="" THEN 8Ø
9Ø IF B$="Q" THEN CLOSE
     : END
1ØØ N=Ø
     : GOTO 12Ø
11Ø OP$="ERROR"
     : BY$=""
     : LN=1
12Ø IF INSTR(1, BY$, "/") THEN BY$=LEFT$(BY$,
     INSTR(1, BY$, "/")-1)
13Ø S$=""
     : N=N+1
     : IF N>13 THEN 8Ø ELSE A$=STRING$(15, 32)
     : MID$(A$, 1, 4)=HEX$(PI)
     : FOR X=1 TO LN
     : GOSUB 46Ø
     : MID$(A$, 4+2*X, 2)=R$
     : NEXT
14Ø BY$=LEFT$(BY$, 4)
     : A$=A$+BY$+" "+OP$
     : IF QQ=2 THEN PI=PI+LN
     : PRINT A$
     : IF ZZ=2 THEN PRINT #-2, A$
     : GOTO 4Ø ELSE 4Ø ELSE PRINT A$; " "; CHR$(95);
     : N=Ø
15Ø Q$=INKEY$
     : IF Q$="" THEN 15Ø ELSE IF Q$="^" OR
     Q$=CHR$(1Ø) THEN PI=PI+LN
     : PRINT CHR$(13);
     : GOTO 4Ø ELSE IF Q$=CHR$(13) THEN 17Ø ELSE IF
     Q$="@" THEN PRINT CHR$(13);
     : GOTO 3Ø ELSE IF Q$="Q" THEN CLOSE
     : END
16Ø S$=S$+Q$
     : PRINT Q$;
     : GOTO 15Ø
17Ø LN=LEN(S$)/2
     : FOR X=1 TO LN
     : POKE PI+X-1, VAL("&H"+MID$(S$, 2*X-1, 2))
     : NEXT X
     : PRINT CHR$(13);
     : GOTO 4Ø
18Ø LN=2+TE
     : OP$=HEX$(PEEK(PI+TE+1))
     : RETURN
19Ø LN=4+TE
     : X=TE
     : TE=TE+1
     : GOSUB 29Ø
     : TE=X
     : GOTO 27Ø
2ØØ BY$=MID$(BY$, X+1)
     : GOTO 6Ø
21Ø LN=1+TE
     : OP$=""
     : RETURN
22Ø BY=PEEK(PI+TE+1)
     : IF BY>127 THEN 23Ø ELSE LN=2+TE
     : GOSUB 47Ø
     : BY=31 AND BY
     : IF BY>15 THEN OP$=R$+"-"+HEX$(32-BY)
     : RETURN ELSE OP$=R$ + "+" + HEX$(BY)
     : RETURN
23Ø GOSUB 47Ø
     : BY=31 AND BY
     : GET #2, ((15 AND BY)+1)
     : OP$=I$
     : GOSUB 5ØØ
24Ø IF LEFT$(OP$, 1)="X" THEN 19Ø ELSE IF LEFT$(OP$,
     1)=" " THEN OP$="ERROR"
     : BY$=""
     : LN=1
     : RETURN ELSE LN=2+TE+VAL(OP$)
     : IF VAL(OP$)=Ø THEN 27Ø ELSE IF VAL(OP$)=2 THEN
     26Ø ELSE OP$=MID$(OP$, 2)
     : BY=PEEK(PI+2+TE)
     : IF BY<128 THEN OP$=OP$ + "+" + HEX$(BY)
     : GOTO 27Ø
25Ø OP$=OP$ + "-" + HEX$(256-BY)
     : GOTO 27Ø
26Ø BY=PEEK(PI+2+TE)*256+PEEK(PI+3+TE)
     : OP$=MID$(OP$, 2)
     : IF BY>32767 THEN OP$=OP$ + "-" +
     HEX$(65536-BY) ELSE OP$=OP$ + "+" + HEX$(BY)
27Ø IF INSTR(1, OP$, "R")<>Ø THEN MID$(OP$, INSTR(1,
     OP$, "R"), 1)=R$
28Ø IF 16 AND PEEK(PI+TE+1) THEN OP$="("+OP$+")"
     : RETURN ELSE RETURN
29Ø LN=3+TE
     : OP$=HEX$(PEEK(PI+TE+1)*256+PEEK(PI+TE+2))
     : RETURN
3ØØ GOSUB 18Ø
     : OP$="#"+OP$
     : RETURN
31Ø GOSUB 29Ø
     : OP$="#" + OP$
     : RETURN
32Ø LN=2
     : BY=PEEK(PI+1)
     : IF BY>127 THEN OP$=HEX$(PI-254+BY)
     : RETURN ELSE OP$=HEX$(PI+2+BY)
     : RETURN
33Ø LN=3+TE
     : BY=PEEK(PI+1+TE)*256+PEEK(PI+2+TE)
     : IF BY>32767 THEN OP$=HEX$(PI+LN-65536+BY)
     : RETURN ELSE OP$=HEX$(PI+LN+BY)
     : RETURN
34Ø LN=2
     : BY=PEEK(PI+1) AND 15
     : GOSUB 44Ø
     : OP$=R$
     : BY=(PEEK(PI+1) AND 24Ø)/16
     : GOSUB44Ø
     : OP$=R$+","+OP$
     : RETURN
35Ø LN=2
     : BY=PEEK(PI+1)
     : OP$=""
     : IF (128 AND BY) THEN OP$="PC"
36Ø IF (64 AND BY) THEN IF MID$(BY$, 4)="U" THEN
     OP$=OP$+",S" ELSE OP$=OP$+",U"
37Ø IF (32 AND BY) THEN OP$=OP$+",Y"
38Ø IF (16 AND BY) THEN OP$=OP$+",X"
39Ø IF (8 AND BY) THEN OP$=OP$+",DP"
```

```
4ØØ IF (4 AND BY) THEN OP$=OP$+",B"
41Ø IF (2 AND BY) THEN OP$=OP$+",A"
42Ø IF (1 AND BY) THEN OP$=OP$+",CC"
43Ø IF MID$(OP$, 1, 1)="," THEN OP$=MID$(OP$, 2)
     : RETURN ELSE RETURN
44Ø IF BY=Ø THEN R$="D" ELSE IF BY=1 THEN R$="X" ELSE
     IF BY=2 THEN R$="Y" ELSE IF BY=3 THEN R$="U"
     ELSE IF BY=4 THEN R$="S" ELSE IF BY=5 THEN
     R$="PC" ELSE IF BY=8 THEN R$="A" ELSE IF BY=9
     THEN R$="B" ELSE IF BY=1Ø THEN R$="CC" ELSE IF
     BY=11 THEN R$="DP" ELSE R$="??"
45Ø RETURN
46Ø R$=HEX$(PEEK(PI+X-1))
     : IF LEN(R$)=1 THEN R$="Ø"+R$
     : RETURN ELSE RETURN
47Ø X=(BY AND 96)/32
     : IF X=Ø THEN R$="X" ELSE IF X=1 THEN R$="Y"
     ELSE IF X=2 THEN R$="U" ELSE IF X=3 THEN R$="S"
48Ø RETURN
5ØØ IF LEN(OP$)=1 THEN RETURN ELSE IF RIGHT$(OP$,
     1)=" " THEN OP$= LEFT$(OP$, LEN(OP$)-1)
     : GOTO 5ØØ
51Ø RETURN
```

TO CREATE SUFFIX.DAT DISK FILE OF 6809 CODE

```
1Ø CLS
     : PRINT " I N I T I A L I Z A T I O N "
2Ø CLEAR 1ØØØ
     : DIM N$(255), I$(15)
     : FOR X=Ø TO 255
     : READ N$(X)
     : NEXT
     : FOR X=Ø TO 15
     : READ I$(X)
     : NEXT
3Ø DATA 1NEG, , , 1COM, 1LSR, , 1ROR, 1ASR
35 DATA 1ASL, 1ROL, 1DEC, , 1INC, 1TST, 1JMP, 1CLR
4Ø DATA P, P, 2NOP, 2SYNC, , , 8LBRA, 8LBSR, , 2DAA
45 DATA 5ORCC, , 5ANDCC, 2SEX, 9EXG, 9TFR
5Ø DATA 7BRA, 7BRN/8LBRN, 7BHI/8LBHI, 7BLS/8LBLS
52 DATA 7BHS/8LBHS, 7BLO/8LBLO, 7BNE/8LBNE,
     7BEQ/8LBEQ
55 DATA 7BVC/8LBVC, 7BVS/8LBVS, 7BPL/8LBPL,
     7BMI/8LBMI
58 DATA 7BGE/8LBGE, 7BLT/8LBLT, 7BGT/8LBGT,
     7BLE/8LBLE
6Ø DATA 3LEAX, 3LEAY, 3LEAS, 3LEAU, ØPSHS, ØPULS,
     ØPSHU
65 DATA ØPULU, , 2RTS, 2ABX, 2RTI, 2CWAI
68 DATA 2MUL, , 2SWI/2SWI2/2SWI3
7Ø DATA 2NEGA, , , 2COMA, 2LSRA, , 2RORA
75 DATA 2ASRA, 2ASLA, 2ROLA, 2DECA, , 2INCA, 2TSTA, ,
     2CLRA
8Ø DATA 2NEGB, , , 2COMB, 2LSRB, , 2RORB
85 DATA 2ASRB, 2ASLB, 2ROLB, 2DECB, , 2INCB, 2TSTB, ,
     2CLRB
9Ø DATA 3NEG, , , 3COM, 3LSR, , 3ROR, 3ASR
95 DATA 3ASL, 3ROL, 3DEC, , 3INC, 3TST, 3JMP, 3CLR
1ØØ DATA 4NEG, , , 4COM, 4LSR, , 4ROR, 4ASR
1Ø5 DATA 4ASL, 4ROL, 4DEC, , 4INC, 4TST, 4JMP, 4CLR
11Ø DATA 5SUBA, 5CMPA, 5SBCA, 6SUBD/6CMPD/6CMPU,
     5ANDA, 5BITA
115 DATA 5LDA, , 5EORA, 5ADCA, 5ORA, 5ADDA,
     6CMPX/6CMPY/6CMPS
116 DATA 7BSR, 6LDX/6LDY, , 1SUBA, 1CMPA, 1SBCA
117 DATA 1SUBD/1CMPD/1CMPU, 1ANDA, 1BITA, 1LDA, 1STA
12Ø DATA 1EORA, 1ADCA, 1ORA, 1ADDA
122 DATA 1CMPX/1CMPY/1CMPS, 1JSR, 1LDX/1LDY,
     1STX/1STY
13Ø DATA 3SUBA, 3CMPA, 3SBCA, 3SUBD/3CMPD/3CMPU
132 DATA 3ANDA, 3BITA, 3LDA, 3STA, 3EORA
135 DATA 3ADCA, 3ORA, 3ADDA, 3CMPX/3CMPY/3CMPS
136 DATA 3JSR, 3LDX/3LDY, 3STX/3STY, 4SUBA, 4CMPA
14Ø DATA 4SBCA, 4SUBD/4CMPD/4CMPU, 4ANDA, 4BITA,
     4LDA, 4STA
145 DATA 4EORA, 4ADCA, 4ORA, 4ADDA
146 DATA 4CMPX/4CMPY/4CMPS, 4JSR, 4LDX/4LDY,
     4STX/4STY
15Ø DATA 5SUBB, 5CMPB, 5SBCB, 6ADDD, 5ANDB, 5BITB
152 DATA 5LDB, , 5EORB, 5ADCB, 5ORB, 5ADDB
153 DATA 6LDD, , 6LDU/6LDS, , 1SUBB, 1CMPB
16Ø DATA 1SBCB, 1ADDD, 1ANDB, 1BITB, 1LDB, 1STB
162 DATA 1EORB, 1ADCB, 1ORB, 1ADDB, 1LDD, 1STD
163 DATA 1LDU/1LDS, 1STU/1STS
17Ø DATA 3SUBB, 3CMPB, 3SBCB, 3ADDD, 3ANDB, 3BITB
172 DATA 3LDB, 3STB, 3EORB, 3ADCB, 3ORB, 3ADDB
173 DATA 3LDD, 3STD, 3LDU/3LDS, 3STU/3STS
18Ø DATA 4SUBB, 4CMPB, 4SBCB, 4ADDD, 4ANDB, 4BITB
182 DATA 4LDB, 4STB, 4EORB, 4ADCB, 4ORB, 4ADDB, 4LDD
184 DATA 4STD, 4LDU/4LDS, 4STU/4STS
19Ø DATA R+, R++, -R, --R, R+Ø, R+ACCB
192 DATA R+ACCA, , 1R, 2R, , R+ACCD, 1PC, 2PC, , X
2ØØ OPEN "D", #1, "CODE.DAT", 18
     : FIELD #1, 18 AS CODE$
21Ø FOR X=Ø TO 255
     : LSET CODE$=N$(X)
     : REC=X+1
     : PUT #1, REC
     : NEXT X
     : CLOSE #1
22Ø OPEN "D", #1, "SUFFIX.DAT", 6
     : FIELD #1, 6 AS SUFFIX$
23Ø FOR X=Ø TO 15
     : LSET SUFFIX$=I$(X)
     : REC=X+1
     : PUT #1, REC
     : NEXT X
     : CLOSE #1
```

Disk Directory

Bery Rinaldo
1860 Robin Hood Avenue
Titusville, FL 32780

I recently received the disk system for my 32K Extended BASIC Color Computer (with Line Printer VII). After saving my first few programs on the disk, I then executed the "DIR" command in order to see what I had on the disk. Seeing this, my brother and I thought it would be nice to have the directories on hard copy for future reference. That conversation inspired me to write this 25-line program which lists name, extension, and length in granules for each program on the disk and also prints the number of free granules left on the disk. I hope this will help other users with the same need/problem.

Thank you for all of your effort in the newsletter, it is well appreciated by all of us users.

```
1ØØ CLEAR 1ØØØ
     : DIM A(67)
11Ø !Ø, 17, 2, B$, A$
12Ø FOR W=1 TO 68
13Ø A(W-1)=ASC(MID$(B$, W, 1))
14Ø NEXT W
15Ø PRINT #-2, TAB(33); "Disk Directory"
     : PRINT #-2
     : Z=Ø
16Ø FOR X=3 TO 11
17Ø !Ø, 17, X, A$(1), A$(2)
18Ø FOR Q=1 TO 2
19Ø FOR I=1 TO 128 STEP 32
2ØØ IF ASC(MID$(A$(Q), I+11, 1))=255 THEN 32Ø
21Ø T$=MID$(A$(Q), I, 8)
22Ø FOR V=1 TO 8
23Ø C=ASC(MID$(T$, V, 1))
24Ø IF C>122 OR C<32 THEN 32Ø
```

```
25Ø NEXT V
26Ø E$=MID$(A$(Q), I+8, 3)
     : H=Ø
27Ø S=ASC(MID$(A$(Q), I+13, 1))
28Ø H=H+1
     : IF A(S)>67 THEN 3ØØ
29Ø S=A(S)
     : GOTO 28Ø
3ØØ PRINT #-2, TAB(Z); T$ "/" E$; STR$(H);
31Ø Z=Z+16
     : IF Z>66 THEN PRINT #-2
     : Z=Ø
32Ø NEXT I, Q, X
33Ø IF Z>Ø THEN PRINT #-2
34Ø PRINT #-2
     : PRINT #-2, "Free granules left on disk =";
     !(Ø)
```

Character Recognition

William Robinson
457 Main Street
Newington, CT 06111

Here is a program for the 4K Color Computer.

```
1 'WILLIAM ROBINSON   1982
5 CLS
1Ø PRINT @169, "DAVIDS PROGRAM"
2Ø FOR X=1 TO 1ØØØ
     : NEXT X
3Ø CLS
31 'RESTORE DATA GET NEXT WORD
32 RESTORE
34 IF A3=Ø THEN 38
36 FOR X=1 TO A3
     : READ A$
     : NEXT
38 A3=A3+1
4Ø READ A$
44 ' Z MUST BE LAST DATA BEFORE SONG DATA
45 IF A$="Z" THEN 25Ø
5Ø FOR X=1 TO LEN(A$)
6Ø PRINT @169+(X*2), MID$(A$, X, 1)
7Ø NEXT X
75 GOSUB 2ØØØ
8Ø FOR X=1 TO LEN(A$)
85 PRINT @2Ø1+(X*2), "^"
87 PRINT @2Ø1+((X-1)*2), " ";
89 GOSUB 2ØØØ
9Ø B$=INKEY$
1ØØ IF B$="" THEN 9Ø
11Ø IF B$=MID$(A$, X, 1) THEN 13Ø
12Ø SOUND 1Ø, 6
     : GOTO9Ø
13Ø FOR Y=1 TO 3
14Ø SOUND 15Ø, 2
15Ø NEXT Y
16Ø PRINT @233+(X*2), B$
17Ø NEXT X
177 GOSUB 2ØØØ
18Ø GOSUB 1ØØØ
185 RESTORE
186 FOR X=1 TO 7
     : READ A$
     : NEXT
187 'LOOP EXIT MUST = # WORDS INCLUDING "Z"
188 FOR X=1 TO 42
189 READ A
19Ø IF X/7=INT(X/7) THEN B=6 ELSE B=3
193 SOUND A, B
195 NEXT X
2ØØ GOTO 5
22Ø DATA "DAVID", "HOUSE", "DOOR", "DEBBIE", "MOMMY",
    "DADDY"
23Ø 'ANY # OF WORDS MAY BE ADDED
231 '"Z" MUST BE LAST
232 'LINE 186 MUST READ # WORDS
233 '"Z" MUST BE BEFORE SONG
24Ø DATA "Z"
25Ø END
26Ø DATA 89, 89, 147, 147, 159, 159
27Ø DATA 147, 133, 133, 125, 125, 1Ø8, 1Ø8
28Ø DATA 89, 147, 147, 133, 133, 125, 125
29Ø DATA 1Ø8, 147, 147, 133, 133, 125, 125
3ØØ DATA 1Ø8, 89, 89, 147, 147, 159, 159
31Ø DATA 147, 133, 133, 125, 125, 1Ø8, 1Ø8, 89
1ØØØ CLS(Ø)
1ØØ5 FOR X=8 TO 54
1Ø1Ø SET (X, 2, 3)
1Ø15 SET(X, 28, 3)
1Ø2Ø NEXT X
1Ø3Ø FOR X=2 TO 28
1Ø4Ø SET(8, X, 3)
1Ø5Ø SET(54, X, 3)
1Ø6Ø NEXT X
1Ø8Ø FOR X=18 TO 23
1Ø9Ø FOR Y=6 TO 9
11ØØ SET (X, Y, 4)
11Ø5 SET (X+2Ø, Y, 4)
111Ø NEXT Y, X
112Ø X=3Ø
     : Y=15
113Ø SET (X+1, Y, 4)
     : SET(X, Y+1, 4)
114Ø SET (X, Y, 4)
     : SET (X+1, Y+1, 4)
1155 Y=293
116Ø FOR X=Ø TO 3
117Ø PRINT @Y+X, CHR$(191);
118Ø PRINT @Y+2Ø-X, CHR$(191);
119Ø Y=Y+32
12ØØ NEXT X
121Ø FOR X=18 TO 44 STEP4
122Ø SET (X, 24, 4)
     : SET (X+1, 24, 4)
123Ø SET (X+2, 25, 4)
     : SET (X+3, 25, 4)
124Ø NEXT X
125Ø RETURN
2ØØØ AUDIO ON
2Ø1Ø MOTOR ON
2Ø2Ø FOR Z4=1 TO 8ØØ
     : NEXT Z4
2Ø3Ø MOTOR OFF
2Ø4Ø AUDIO OFF
2Ø5Ø RETURN
```

Routine to load text:

```
1Ø 'ROUTINE TO LOAD TEXT
2Ø 'BILL ROBINSON
3Ø '457 MAIN ST.
4Ø 'NEWINGTON CONN. Ø6111
5Ø OPEN "I", #-1, "TEXT"
55 CLS
6Ø FOR X=1 TO 15
7Ø IF EOF(-1) THEN 1ØØ
8Ø INPUT #-1, A2$
9Ø PRINT  MID$(A2$, 2)
95 NEXT
96 PRINT "press enter to cont";
97 INPUT S$
98 GOTO 55
1ØØ END
45Ø OPEN "O", #-1, "TEXT"
46Ø CLS
     : PRINT "BEGIN TEXT"
495 REM FILL THIS LINE
497 A2$="/"
```

```
498 PRINT "/";
5ØØ FOR X=1 TO  31
51Ø A$=INKEY$
512 IF A$="" THEN 51Ø
513 PRINT A$;
515 IF A$=CHR$(13)THEN X=35
     : GOTO 54Ø
52Ø IF A$=CHR$(1Ø) THEN 68Ø
525 IF X=27 THEN SOUND 175, 5
532 IF A$=CHR$(8) THEN GOTO 69Ø
535 A2$=A2$+A$
54Ø NEXT
55Ø PRINT #-1, A2$
67Ø GOTO 495
68Ø END
69Ø Q=LEN(A2$)
7ØØ A2$=LEFT$(A2$, Q-1)
71Ø X=X-1
     : IF X<1 THEN X=1
72Ø GOTO 51Ø
```

Notepad Calculator For the Color Computer

ABEC James McCracken
Abe "C" School
Nattc Lakehurst, N.J. 08733

The program runs as follows, the first input is the operation to be performed, next are the numbers being worked with. The program then supplies the answer and asks if the answer is to be saved. If the answer is to be saved, the program asks in what position, and asks for a name for the value. All this information is then displayed at the bottom of the screen. To recall numbers in the memory, simply enter a 0 for a number prompt. At this point the program asks which number, simply enter the variable number ei. (A(1) = 1) the program will then display the value and return to the main routine.

```
1Ø '**** NOTE PAD ****
2Ø '**** BY ****
3Ø '**** JIM MCCRACKEN ****
4Ø CLS
     : PRINT @ 11, "NOTE PAD"
5Ø PRINT @ 32Ø, "A(1)"
     : PRINT @ 352, "A(2)"
     : PRINT @ 384, "A(3)"
     : PRINT @ 416, "A(4)"
     : PRINT @ 448, "A(5)"
     : PRINT @ 48Ø, "A(6)";
6Ø PRINT @ 324, A$(1)
     : PRINT @ 356, A$(2)
     : PRINT @ 388, A$(3)
     : PRINT @ 42Ø, A$(4)
     : PRINT @ 452, A$(5)
     : PRINT @ 484, A$(6);
7Ø PRINT @ 332, A(1)
     : PRINT @ 364, A(2)
     : PRINT @ 396, A(3)
     : PRINT @ 428, A(4)
     : PRINT @ 46Ø, A(5)
     : PRINT @ 492, A(6);
8Ø PRINT @64, "(+,-,/,*) WHAT ARE WE GOING TO DO";
9Ø INPUT B$
1ØØ INPUT " FIRST #"; C1
11Ø IF C1=Ø GOTO 28Ø
12Ø INPUT "SECOND #"; C2
13Ø IF C2=Ø GOTO 32Ø
14Ø IF B$="+" THEN 18Ø ELSE IF B$="-" THEN 19Ø ELSE
     IF B$="/" THEN 2ØØ ELSE IF B$="*" THEN 21Ø ELSE
     22Ø
15Ø PRINT "ANSWER " C3
16Ø PRINT @ 64, "";
     : INPUT "DO YOU WANT TO SAVE THIS ANS.Y, N"; D$
17Ø IF D$="Y" GOTO 23Ø ELSE GOTO 1Ø
18Ø C3=C1+C2
     : GOTO 15Ø
19Ø C3=C1-C2
     : GOTO 15Ø
2ØØ C3=C1/C2
     : GOTO 15Ø
21Ø C3=C1*C2
     : GOTO 15Ø
22Ø PRINT "I DON'T UNDERSTAND " B$ " REDO";
     : INPUT B$
     : GOTO 14Ø
23Ø '**** SAVING ROUTINE ****
24Ø INPUT "UNDER WHICH POSITION (A1-6)"; E
25Ø INPUT "WHAT NAME"; A$(E)
26Ø A(E)=C3
27Ø GOTO 1Ø
28Ø PRINT @ 128, "";
     : INPUT "INPUT VAR #"; V
29Ø C1=A(V)
3ØØ PRINT @128, " FIRST #" C1
31Ø GOTO 12Ø
32Ø PRINT @16Ø, "";
     : INPUT "INPUT VAR #"; V
33Ø C2=A(V)
34Ø PRINT @16Ø, "SECOND #" C2
35Ø GOTO 14Ø
```

Homebudget

Richard G. Neill
5170 Bramblewood Dr.
Acworth, GA 30101

The following is a home budget program. It is in BASIC, for the Color Computer, and requires a printer. The program asks for your budget period, income, and expenses. There are several "fixed" expense entries and then 28 "variable expense" categories. As you input the values for the income and fixed expenses they are tabulated inside the program.

The variable expenses are entered and then proportioned so that their total equals the "spendable" income.

You are allowed to go through the list as many times as you like. After the first time through, as each category is presented, you are asked whether to change it and whether to freeze it. If you freeze it, you can not change it on the next pass through. If you do not input a value for an expense, when it is first presented, it will be deleted from subsequent passes. After each section is completed, a summary is presented on the video. When the variable expense summary is presented, any frozen accounts are off-set to the right. After all accounts are frozen, you are given the option of a hard copy.

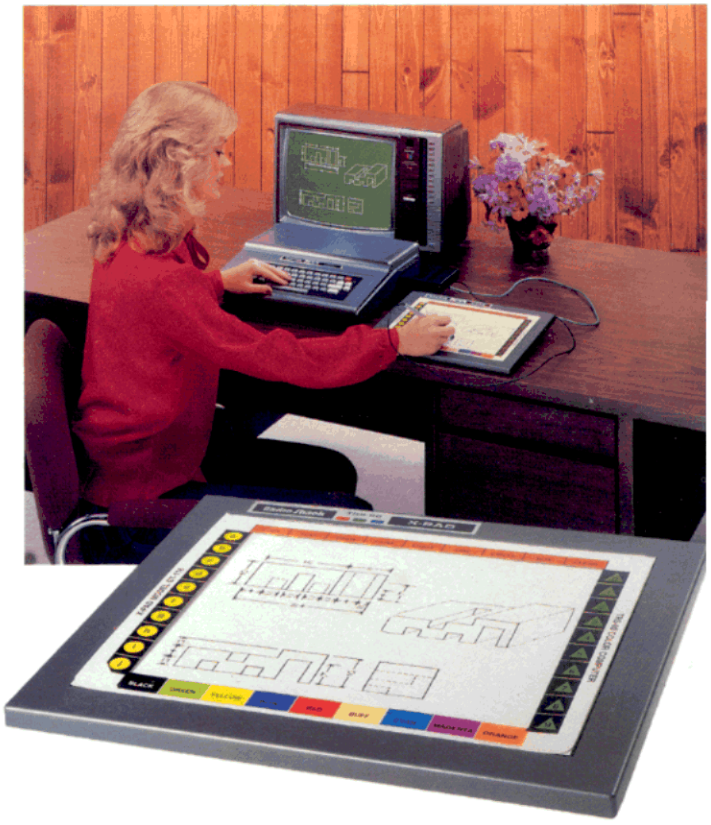
If you select a hard copy your name is printed at the top and the report is then printed.

```
1Ø '*******************************
2Ø '******** HOME BUDGET *********
3Ø '***** GARRY NEILL,MAR 82 *****
4Ø '********** REV 1.Ø ***********
5Ø '***** THANKS, BILL DIAL ******
6Ø PMODE Ø
     : PCLEAR 1
7Ø Q=27
```

```
     : V$="###.#"
     : W$="$$#####,"
     : U$="###"
8Ø CLS
9Ø DIMD(18), E$(Q), V(Q), F(Q)
1ØØ PRINT TAB(7) "HOME BUDGETING MODEL"
     : PRINT
11Ø DATA SALARY/WAGES, OTHER INCOME, FED INC TAX,
     STATE & LOCAL TAX
12Ø DATA SOCIAL SECURITY, UNEMPLOYMENT INS, HEALTH
     INS
13Ø DATA LIFE INS, CONTRIBUTIONS, OTHER DEDUCTIONS
14Ø DATA RENT/MORTGAGE. LIFE INS, HEALTH INS, HOUSE
     INS
15Ø DATA AUTO INS, CAR PAYMENTS, LOAN PAYMENTS, TRASH
     REMOVAL
16Ø DATA OTHER FIXED EXP
17Ø DATA FOOD, CLOTHING, DRY CLEANING, BARBER
18Ø DATA HOME MTC, NATURAL GAS, WATER, ELECTRICITY,
     TELEPHONE
19Ø DATA GAS/OIL, AUTO MTC, PARKING, DENTIST
2ØØ DATA DOCTOR, DRUGS, SCHOOL, ALLOWANCE
21Ø DATA CLUBS, LODGES, SPORTS, RESTURANTS
22Ø DATA ENTERTAINMENT, BOOKS, SITTERS, NURSERY
23Ø DATA VACATION SAVINGS, MISC SAVINGS, TITHE, MISC.
24Ø PRINT
     : PRINT "SELECT YOUR BUDGETING PERIOD BY NUMBER.
     LATER ON IT WILL BE"
25Ø PRINT "EXTENDED TO ONE YEAR."
     : PRINT
26Ø PRINT "1- WEEKLY"
     : PRINT "2- BIWEEKLY"
     : PRINT "3- SEMIMONTHLY"
     : PRINT "4- MONTHLY"
     : PRINT
27Ø INPUT P
     : IF P>4 THEN PRINT "TRY AGAIN"
     : GOTO 27Ø
28Ø IF P=1 THEN P=52 ELSE  IF P=2 THEN P=26 ELSE IF
     P=3 THEN  P=24 ELSE IF P=4 THEN P=12
29Ø PRINT "ALRIGHT, FIRST LET'S LOOK AT    INCOME FOR
     THE PERIOD."
3ØØ READ A$
     : PRINT A$; " $";
     : INPUT D(Ø)
     : READ A$
     : PRINT A$; " $";
     : INPUT D(1)
31Ø TI=D(Ø)+D(1)
     : PRINT
32Ø PRINT
     : PRINT "OK, NOW LET'S LOOK AT PAYCHECK
     DEDUCTIONS."
     : PRINT
33Ø FOR J=2 TO 9
     : READ A$
     : PRINT A$;
     : INPUT " $"; D(J)
     : TD=TD+D(J)
     : NEXT J
     : PRINT
34Ø PRINT "OK, NOW LET'S LOOK AT FIXED EX- PENSES."
35Ø FOR J=1Ø TO 18
     : READ A$
     : PRINT A$;
     : INPUT " $"; D(J)
     : TF=TF+D(J)
     : NEXT J
36Ø DF=TD+TF
     : SI=TI-DF
     : S=32
37Ø PRINT
     : PRINT "OK, AT THIS TIME OUR TABLE LOOKS LIKE
     THIS:"
38Ø '
39Ø '
4ØØ PRINT "        HOME BUDGETING MODEL"
41Ø PRINT "   SPENDABLE INCOME SUMMARY"
42Ø GOSUB 189Ø
     : PRINT TAB(3) "ACCT." TAB(2Ø) "PER." TAB(26)
     "ANNUAL";
43Ø GOSUB 189Ø
     : PRINT "TOTAL INCOME" TAB(16);
     : PRINT USING W$; TI;
     : PRINT TAB(18)
44Ø PRINT USING W$; TI*P
     : PRINT " DED." TAB(8);
45Ø PRINT USING W$; TD*(-1)
46Ø PRINT "FIX.EXP." TAB(4);
     : PRINT USING W$; TF*(-1);
47Ø PRINT TAB(1Ø);
     : PRINT USING W$; DF*(-1);
     : PRINT TAB(5);
48Ø PRINT USING W$; DF*(-1)*P;
49Ø PRINT TAB(11) "------" TAB(19) "-----" TAB(25)
     "-------"
     : PRINT "SPENDABLE INCOME";
5ØØ PRINT TAB(16);
     : PRINT USING W$; SI;
     : PRINT TAB(5);
     : PRINT USING W$; SI*P;
51Ø PRINT TAB(19) "=====" TAB(25) "======="
52Ø '
53Ø '
54Ø GOSUB 191Ø
     : RO=Ø
55Ø CLS
     : PRINT "OK, NOW FOR THE FIRST ROUND OF
     VARIABLE EXPENSE. DON'T PINCH"
56Ø PRINT "YOURSELF IN YOUR ESTIMATES"
57Ø PRINT "(WITHIN REASON). LET THE"
58Ø PRINT "COMPUTER HELP YOU REFINE YOUR   BUDGET
     LATER ON."
     : PRINT
59Ø FOR J=Ø TO Q
     : READ E$(J)
     : PRINT E$(J);
     : INPUT " $"; V(J)
     : VT=VT+V(J)
     : NEXT J
6ØØ RESTORE
     : PRINT
     : PRINT
61Ø PRINT "YOUR BUDGET FOR THE FIRST       ROUND
     TOTALED $" VT ", THIS"
62Ø PRINT "COMPARES TO SPENDABLE INCOME    OF $" SI
     ". WE HAVE PRORATED"
63Ø PRINT "THE DIFFERENCE, $" SI-VT ", OVER"
64Ø PRINT "ALL VARIABLE EXPENSE ACCOUNTS."
65Ø GOSUB 191Ø
     : FOR J=Ø TO Q
     : V(J)=INT(V(J)/VT*SI)
     : NEXT
     : VT=SI
     : PRINT
66Ø PRINT "NOW WE BEGIN THE BUDGET REFINE- MENT
     PHASE. MAKE AS MANY"
67Ø PRINT "PASSES AS YOU LIKE. AS YOU"
68Ø PRINT "REVIEW EACH ACCOUNT, DECIDE"
69Ø PRINT "WHETHER TO FREEZE IT OR TO       LEAVE IT
     FOR ANOTHER PASS."
     : PRINT
7ØØ PRINT "HINT: DON'T BE IN A HURRY TO    FREEZE AN
     ACCOUNT."
     : PRINT
71Ø PRINT "YOUR TASK IS FINISHED WHEN ALL  ACCOUNTS
     ARE FROZEN."
     : PRINT
72Ø GOSUB 191Ø
73Ø FOR J=Ø TO Q
     : PRINT
     : IF V(J)=Ø THEN 93Ø
74Ø PRINT E$(J); " $"; V(J)
     : INPUT "CHANGE Y/N"; A$
```

```
75Ø IF LEFT$(A$, 1)="N" THEN 79Ø ELSE IF LEFT$(A$,
     1)="Y" THEN 77Ø ELSE IF LEFT$(A$, 1)<>"Y" THEN
     74Ø
76Ø GOTO 74Ø
77Ø INPUT "REVISED AMOUNT $"; A
     : IF A<V(J) THEN 79Ø
78Ø GOSUB 193Ø
79Ø INPUT "FREEZE Y/N"; B$
     : PRINT
8ØØ IF LEFT$(A$, 1)="Y" AND LEFT$(B$, 1)="Y" THEN 85Ø
81Ø IF LEFT$(A$, 1)="Y" AND LEFT$(B$, 1)="N" THEN 89Ø
82Ø IF LEFT$(A$, 1)="N" AND LEFT$(B$, 1)="N" THEN 93Ø
83Ø VT=VT-V(J)
     : FT=FT+V(J)
     : C=F(J)
     : F(J)=V(J)
     : V(J)=C
     : V(J)=Ø
     : PRINT
84Ø PRINT "OK, THE ACCOUNT HAS BEEN FROZEN  AT $"
     F(J) "."
     : GOTO 93Ø
85Ø VT=VT-A
     : FT=FT+A
     : C=A
     : A=F(J)
     : F(J)=C
     : V(J)=Ø
86Ø FOR K=Ø TO Q
     : V(K)=INT(V(K)*(SI-FT)/VT)
     : NEXT
     : VT=SI-FT
87Ø PRINT "OK, THE ACCOUNT HAS BEEN FROZEN AT $" F(J)
     ", AND THE DIFFERENCE"
88Ø PRINT "PRORATED OVER REMAINING ACOUNTS."
     : PRINT
     : GOTO 93Ø
89Ø VT=VT-V(J)+A
     : V(J)=A
9ØØ FOR K=Ø TO Q
     : V(K)=INT(V(K)*(SI-FT)/VT)
     : NEXT
     : VT=SI-FT
91Ø PRINT "OK, THE VALUE HAS BEEN CHANGED  TO $" A ",
     AND THE DIFFERENCE"
92Ø PRINT "PRORATED OVER REMAINING ACOUNTS"
     : PRINT
93Ø NEXT J
     : PRINT
     : R=Ø
     : FOR J=Ø TO  Q
     : IF V(J)=Ø THEN R=R+1
     : NEXT
94Ø IF R<=Q THEN 96Ø
95Ø FOR J=Ø TO Q
     : F(J)=INT(F(J)/FT*SI)
     : VT=Ø
     : FT=SI
     : GOTO 134Ø
96Ø PRINT "NOW LET'S RECAP VARIABLE        EXPENSES
     BEFORE GOING ON TO THE"
97Ø PRINT "NEXT ROUND."
     : PRINT
     : RO=RO+1
     : V1=Ø
     : F1=Ø
98Ø FOR J=Ø TO  Q
     : V(J)=INT(V(J)/VT*(SI-FT))
     : NEXT
     : VT=SI-FT
99Ø '
1ØØØ '
1Ø1Ø PRINT "INT. SUM. OF VAR. EXP.";
1Ø2Ø PRINT ":  NO." RO
1Ø3Ø GOSUB 189Ø
     : PRINT " ACTS." TAB(8) "NUM." TAB(15) "%"
     TAB(2Ø) "AMT" TAB(3Ø) "%"
1Ø4Ø GOSUB 189Ø
     : FOR J=Ø TO Q
     : IF V(J)+F(J)=Ø THEN 1Ø6Ø
1Ø5Ø IF V(J) THEN V1=V1+1 ELSE F1=F1+1
1Ø6Ø NEXT J
1Ø7Ø PRINT "VAR EXP" TAB(6);
     : PRINT USING U$; V1;
     : PRINT TAB(11);
1Ø8Ø PRINT USING V$; V1/(V1+F1)*1ØØ;
     : PRINT TAB(15);
     : PRINT USING W$; VT;
1Ø9Ø PRINT TAB(26);
     : PRINT USING V$; VT/TI*1ØØ
11ØØ PRINT "FZN EXP" TAB(6);
     : PRINT USING U$; F1;
     : PRINT TAB(11);
111Ø PRINT USING V$; F1/(V1+F1)*1ØØ;
     : PRINT TAB(15);
112Ø PRINT USING W$; FT;
     : PRINT TAB(26);
     : PRINT USING V$; FT/TI*1ØØ
113Ø PRINT TAB(12) "------" TAB(24) "------"
114Ø PRINT "TOTALS" TAB(7);
     : PRINT USING U$; F1+V1;
     : PRINT TAB(16);
115Ø PRINT USING W$; VT+FT
116Ø GOSUB 191Ø
117Ø GOSUB 118Ø
     : GOTO 122Ø
118Ø PRINT "INTERIM VAR. EXPENSES NO." RO
119Ø GOSUB 189Ø
     : PRINT "  ACT." TAB(16) "VAR." TAB(22)
     "FROZEN";
12ØØ PRINT TAB(28) "%"
121Ø GOSUB 189Ø
122Ø FOR J=Ø TO Q
     : IF V(J)+F(J)=Ø THEN 129Ø ELSE PRINT E$(J);
     : R2=R2+1
123Ø IF F(J) THEN 125Ø ELSE PRINT TAB(8)
     : PRINT USING W$; V(J);
     : PRINT TAB(21);
124Ø PRINT USING V$; V(J)/TI*1ØØ
     : GOTO 127Ø
125Ø PRINT TAB(16);
     : PRINT USING W$; F(J);
     : PRINT TAB(22);
126Ø PRINT USING V$; F(J)/TI*1ØØ
127Ø IF INT(R2/5)=R2/5  THEN PRINT
128Ø G=G+1
     : IF G=>11 THEN GOSUB 191Ø
     : G=Ø
129Ø NEXT J
     : R2=Ø
13ØØ '
131Ø '
132Ø PRINT "NOW WE MAKE ANOTHER PASS THROUGH VARIABLE
     EXPENSES."
     : PRINT
133Ø GOTO 72Ø
134Ø INPUT "HARDCOPY Y/N"; A$
     : IF LEFT$(A$, 1)="Y" THEN 135Ø ELSE 2ØØØ
135Ø IF PEEK(65314)<>4 THEN PRINT "PRINTER BUSY" ELSE
     137Ø
136Ø GOTO 134Ø
137Ø PRINT "PLEASE INPUT THE REQUESTED
     INFORMATION BEFORE WE PROVIDE A"
138Ø PRINT "FINAL DETAIL OF YOUR BUDGET."
     : PRINT
139Ø LINEINPUT "NAME  "; A$
     : INPUT "BUDGET YEAR"; A
14ØØ B=LEN(A$)
     : B=3Ø-B/2
141Ø PRINT #-2, CHR$(27) CHR$(19)
     : PRINT #-2, ""
142Ø '
143Ø GOSUB 192Ø
144Ø GOSUB 19ØØ
```

```
     : PRINT #-2, ""
1450 PRINT #-2, TAB(8) "THE " A$ " BUDGET FOR" A
1460 PRINT #-2, ""
     : PRINT #-2, ""
1470 PRINT #-2, STRING$(80, "*")
1480 PRINT #-2, TAB(6) "ACCOUNT" TAB(20) ":" TAB(26)
     "INCOME" TAB(38) ":";
1490 PRINT #-2, TAB(44) "EXPENSE" TAB(56) ": PERCENT"
1500 PRINT #-2, TAB(20)
     ":-----------------:-----------------: TOTAL"
1510 PRINT #-2, TAB(20) ": PERIOD : ANNUAL : PERIOD :
     ANNUAL : INCOME"
1520 PRINT #-2,
     "--------------------:--------:--------:--------:
     --------:--------"
1530 X$=":$$#####,:$$#####,:         :         : ###.#"
1540 Y$=":         :         :$$#####,:$$#####,: ###.#"
1550 GOTO 1570
1560 PRINT #-2, ":         :         :         :
     :"
     : RETURN
1570 PRINT #-2, "INCOME" TAB(20);
     : PRINT #-2, USING X$; TI; TI*P; TI/TI*100
1580 PRINT #-2, "------" TAB(20);
1590 GOSUB 1560
1600 READ A$
     : PRINT #-2, A$ TAB(20);
     : PRINT #-2, USING X$; D(0); D(0)*P; D(0)/TI*100
1610 READ A$
     : PRINT #-2, A$ TAB(20);
     : PRINT #-2, USING X$; D(1); D(1)*P; D(1)/TI*100
1620 PRINT #-2, TAB(20);
1630 GOSUB 1560
1640 PRINT #-2, "PAYCHECK DEDUCT'S" TAB(20);
1650 PRINT #-2, USING Y$; TD; TD*P; TD/TI*100
1660 PRINT #-2, "-----------------" TAB(20);
1670 GOSUB 1560
1680 FOR J=2 TO 9
     : READ A$
     : IF D(J)=0 THEN 1700  ELSE PRINT #-2, A$
     TAB(20);
1690 PRINT #-2, USING Y$; D(J); D(J)*P; D(J)/TI*100
1700 NEXT J
     : PRINT #-2, TAB(20);
1710 GOSUB 1560
1720 PRINT #-2, "FIXED EXP." TAB(20);
1730 PRINT #-2, USING Y$; TF; TF*P; TF/TI*100
1740 PRINT #-2, "--------------" TAB(20);
1750 GOSUB 1560
1760 FOR J=10 TO 18
     : READ A$
     : IF D(J)=0 THEN 1780 ELSE  PRINT #-2, A$
     TAB(20);
1770 PRINT #-2, USING Y$; D(J); D(J)*P; D(J)/TI*100
1780 NEXT
     : PRINT #-2, TAB(20);
1790 GOSUB 1560
1800 PRINT #-2, "VARIABLE EXPENSE" TAB(20);
1810 PRINT #-2, USING Y$; FT; FT*P; FT/TI*100
1820 PRINT #-2, "----------------" TAB(20);
1830 GOSUB 1560
1840 FOR J=0 TO Q
     : IF F(J)=0 THEN 1860 ELSE PRINT #-2, E$(J)
     TAB(20);
1850 PRINT #-2, USING Y$; F(J); F(J)*P; F(J)/TI*100
1860 NEXT
1870 GOSUB 1900
1880 GOSUB 1900
     : GOTO 2010
1890 FOR L=1 TO S
     : PRINT "-";
     : NEXT
     : RETURN
1900 PRINT #-2, STRING$(80, "=")
     : RETURN
1910 PRINT "PRESS ENTER TO CONTINUE"
     : INPUT A$
     : RETURN
1920 FOR L=1 TO 10
     : PRINT;
     : NEXT
     : RETURN
1930 IF A<=V(J) THEN 1970
1940 IF A<=VT THEN 1970
1950 PRINT "YOU HAVE GONE BUST BY $" A-SI ;
1960 PRINT " *"
     : INPUT "REVISED AMOUNT $"; A
1970 RETURN
1980 RESTORE
     : READ A$
     : READ A$
     : PRINT
     : PRINT
     : S=95
1990 RETURN
2000 GOSUB 1910
2010 CLS
2020 PRINT "WELL THE BUDGET IS FINISHED."
2030 PRINT "NOW FOR A CONSOLIDATED LOOK AT"
2040 PRINT "THE RESULTS."
     : GOSUB 1910
2050 CLS
     : PRINT "HERE ARE YOUR ANNUAL FIGURES"
2060 X$="$$##,###.##"
2070 PRINT "TOTAL INCOME";
     : PRINT TAB(21);
     : PRINT USING X$; TI*P
2080 PRINT "PAYCHECK DEDUCTS";
     : PRINT TAB(21);
     : PRINT USING X$; TD*P
2090 PRINT "TOTAL FIXED EXPENSES";
     : PRINT TAB(21);
     : PRINT USING X$; TF*P
2100 PRINT "TOTAL VAR EXPENSES";
     : PRINT TAB(21);
     : PRINT USING X$; FT*P
2110 PRINT "HOPE YOU'RE HAPPY WITH IT"
```



Radio Shack Computer Center Addresses and Telephone Numbers

ALABAMA

BIRMINGHAM 2428 Green Springs Hwy., (205) 945-0792; 9233 Parkway East, (205) 836-5512
HUNTSVILLE 1400 N. Memorial Pkwy., (205) 536-1581
MOBILE 405 Bel-Air Blvd., (205) 471-1617
MONTGOMERY #24 Union Square S/C, (205) 271-1500

ARIZONA

PHOENIX 10233 Metro Pkwy. E., (602) 861-1124; 4301 N. 7th Ave., (602) 277-3031
SCOTTSDALE 2525 N. Scottsdale Rd., (602) 990-2241
TEMPE 83 E. Broadway, (602) 894-2065
TUCSON 5622 E. Broadway, (602) 748-0101; Campbell Plaza, 2830 N. Campbell Ave., (602) 881-8003

ARKANSAS

LITTLE ROCK Town & Country S/C, University & Asher, (501) 568-5094

CALIFORNIA

ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
BREA Imperial Shopping Center, 391 South State College, (714) 671-1681
CANOGA PARK 8371 Topanga Canyon, (213) 347-9800
CARMICHAEL 6305 Fairoaks Blvd., (916) 484-6815
CHICO 1834 Mangrove Ave., (916) 894-0101
CHULA VISTA 1201 3rd Ave., (619) 420-3810
CITRUS HEIGHTS 7405 Green Back Ln. at San Juan, (916) 723-5183, 969-8752
DOWNEY 8031 Florence Ave., (213) 927-7882
EL TORO El Toro Rd. at San Diego Freeway, (714) 859-4411
ESCONDIDO 347 W. Mission Ave., (619) 741-6032
FREMONT Fremont Hub, 39114 Fremont, (415) 790-1112
FRESNO Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551
GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 24784 Hesperian Blvd., (415) 887-6420
HOLLYWOOD 6922 Hollywood Blvd., (213) 466-8681
IRVINE Redhill at Main St.,(714) 660-0299
LAKEWOOD 5830 Lakewood Blvd., (213) 920-9671
LA MESA 5345 Jackson Dr., (619) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 629-2455; 5240 Century Blvd. (Airport area), (213) 641-4585
MODESTO 221 McHenry Ave., (209) 521-1011
MONTCLAIR 5237 Arrow Hwy., (714) 625-1541
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1933 El Camino Real W., (415) 961-0542
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
PLEASANT HILL 508 G. Contra Costa Blvd., (415) 676-9866
RIVERSIDE 3844 La Sierra Ave., (714) 689-0340
SACRAMENTO 4749 J. St., (916) 454-3287
SAN BERNARDINO 764 Inland Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairemont Dr., (619) 276-6050; 3902 El Cajon Blvd., (619) 280-8027
SAN FRANCISCO One Market Place, (415) 777-9810, 2920 Geary Blvd., (415) 221-8716
SAN JOSE 1228 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA ANA 2320 S. Fairview St. (at Warner), (714) 966-2787
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702
SANTA MONICA 511 Wilshire Blvd., (213) 458-1802
SANTA ROSA 823 4th St., (707) 542-0402
SHERMAN OAKS 14936 Ventura Blvd., (213) 788-2639
STOCKTON College Sq. S/C, 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3840 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

COLORADO

BOULDER Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
COLORADO SPRINGS 4341 N. Academy, (303) 593-7500
DENVER 8000 E. Quincy, (303) 770-1362; Green Briar Plaza, 7075 Pecos, (303) 428-6787
LAKEWOOD 2099 Wadsworth Blvd., (303) 232-6277

CONNECTICUT

EAST HAVEN 51 Frontage Rd., (203) 467-8864
FAIRFIELD 1196 Kings Hwy. & Rt. 1, (203) 255-6099
HARTFORD The Richardson Bldg., 942 Main St., (203) 549-5717
MANCHESTER 228 Spencer St., (203) 649-8210
NORWALK Rt. 7-345 Main Ave., (203) 846-3418
ORANGE Woolco S/C, 538 Boston Post Rd., (203) 795-1291
WATERBURY 105 Bank St., (203) 573-8800
WATERFORD 122 Boston Post Rd., (203) 443-0716
WEST HARTFORD 39 S. Main St., (203) 523-4283

DELAWARE

DOVER Edgehill Shop. Ctr., Rt. #13, (302) 734-5223
WILMINGTON 3847 Kirkwood Hwy., (302) 999-0193

DISTRICT OF COLUMBIA

WASHINGTON 1800 M St. NW., (202) 822-3933; 15th & L St., N.W., (202) 296-2460

FLORIDA

ALTAMONTE SPRINGS 766 E. East Altamonte Dr., (305) 339-1313
BOCA RATON 1662 N. Federal Hwy., (305) 391-3308
CLEARWATER 2460-D US 19 North, (813) 797-3223
DAYTONA BEACH Volusia Plaza, (904) 255-5909
FT. LAUDERDALE 4368 N. Federal Hwy., (305) 772-6797
FORT MYERS Edison Mall, (813) 939-3454
FORT PIERCE Center West Shopping Center, U.S. Hwy. 1, (305) 466-2200
GAINESVILLE 3315 Archer Rd., (904) 375-7063
HOLLYWOOD 429 S. State Rd. #7, (305) 966-4382
JACKSONVILLE 8252 Arlington Expnwy., (904) 725-2594; Roosevelt Mall, Roosevelt Blvd., (904) 388-6611
LAUDERDALE LAKES 4317-25 N. State Rd. 7, (305) 486-2240
MIAMI 9459 S. Dixie Hwy., (305) 667-2316; 1601 Biscayne Blvd. (305) 374-6433; 15 SE. 2nd Ave., (305) 374-7310; 20761 S. Dixie Hwy., (305) 238-2518
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St., (305) 940-6887
ORLANDO 1238 E. Colonial Dr., (305) 894-0570
PENSACOLA Eastgate Shopping Center, 6925 N. 9th Ave., (904) 478-7109
ST. PETERSBURG 3451 66th St. N., (813) 381-2366
SARASOTA 5251 S. Tamiami Tr. (Hwy 41), (813) 923-4721
TALLAHASSEE 2529 S. Adams, (904) 222-4440
TAMPA 4555 W. Kennedy, (813) 879-7470; 1825 E. Fowler Ave., (813) 971-1130
W. PALM BEACH 2271-A Palm Beach Lakes Blvd., (305) 683-3100

GEORGIA

AUGUSTA 3435 Wrightsboro Rd., (404) 738-5998
ATLANTA 2108 Henderson Mill, (404) 939-9888; 49 W. Paces Ferry, (404) 231-9604; Akers Mill S/C, 2937 Cobb Parkway NW., (404) 955-5235; 113 Peachtree St., (404) 223-5904
COLLEGE PARK 5309 Old National Hwy., (404) 761-3055
COLUMBUS Columbus Square Mall, 3060 Macon Rd., (404) 568-7200
DORAVILLE 5697 Buford Hwy., (404) 45?-2691
SAVANNAH Chatham Plaza, 7500 Abercorn St., (912) 355-6074

IDAHO

BOISE 691 S. Capitol Blvd., (208) 344-5450

ILLINOIS

AURORA 890 North Lake St., (312) 844-2224
CHICAGO 4355 S. Archer Ave., (312) 376-7617; CNA Plaza 329 S. Wabash, (312) 922-0536; 72 West Adams, (312) 236-7262; 524 N. Michigan Ave., (312) 822-0533; 101 W. Washington St. at Clark, (312) 782-7207
ELMWOOD PARK 7212 W. Grand Ave., (312) 452-7500
FAIRVIEW HEIGHTS #4 Market Place, (618) 398-6410
HOMEWOOD/GLENWOOD 329 Glenwood Lansing, (312) 758-0440
JOLIET 2415 W. Jefferson St., (815) 744-7080
LaGRANGE One S. LaGrange Rd., (312) 482-3484
LIBERTYVILLE 1350 S. Milwaukee Ave., (312) 367-8230
LOMBARD 4 Yorktown Center, (312) 629-5350
NILES 8349 Golf Rd., (312) 470-0670
OAK LAWN 4815 W. 95th St., (312) 425-9130
PEORIA 4125 N. Sheridan Rd., (309) 685-7056
ROCKFORD North Town S/C, 3600 N. Main St., (815) 282-1001
SCHAUMBURG 651 Mall Dr., (312) 884-8600
SPRINGFIELD Sherwood Plaza, 2478 Wabash, (217) 787-3066

INDIANA

EVANSVILLE 431 Diamond Ave., (812) 426-1715
FT. WAYNE 747 Northcrest S/C, (219) 482-9547
GRIFFITH 208 W. Ridge Rd., (219) 838-3000
HOBART Save-More Plaza, Rtes. #6 & 51, (219) 942-0541
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., (317) 849-6896; Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221; 10013 E. Washington St., (317) 898-4887; Two W. Washington St., (317) 236-9911
SOUTH BEND 1827 South Bend Ave., (219) 277-3789
TERRE HAUTE 3460 U.S. Hwy. 41 S., (812) 234-3212

IOWA

CEDAR RAPIDS 111 First Ave., S.E. (Downtown), (319) 362-7399
DAVENPORT 616 E. Kimberly Rd., (319) 386-3457
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M, (515) 270-0193

KANSAS

FLORENCE 7727 Mall Rd., (606) 371-2811
LEXINGTON 2909 Richmond Rd., (606) 269-7321
LOUISVILLE 2900 Taylorsville Rd., (502) 459-9901; Louisville Galleria, (502) 589-1013

LOUISIANA

ALEXANDRIA 1213 Texas Ave., (318) 473-0429
BATON ROUGE 7007 Florida Blvd., (504) 928-5260
HOUMA 2348 W. Park Ave. (Hwy. 24), (504) 873-7713
LAFAYETTE University Square at Congress Blvd., (318) 235-6177
METAIRIE 3750 Veterans Hwy., (504) 454-3681
NEW ORLEANS 327 St. Charles Ave., (504) 523-6408
SHREVEPORT 1545 Line Ave., (318) 221-5125

MAINE

BANGOR Maine Square, (207) 945-6491

MARYLAND

BALTIMORE 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583; 115 N. Charles St. at Lexington, (301) 539-7251
BETHESDA 7900 Wisconsin Ave., (301) 656-0892
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike, (301) 788-3277
FREDERICK Shoppers World, Rt. 40W, (301) 695-8440
NEW CARROLLTON-LANHAM 7949 Annapolis Rd., (301) 459-8030
PASADENA 8120 Ritchie Hwy., (301) 544-2352
ROCKVILLE Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
SALISBURY Shoppers World S/C, Rt. 50, (301) 546-9223
TEMPLE HILLS 4520 St. Barnabas Rd., (301) 899-0740
TOWSON-LUTHERVILLE Yorktown S/C York Rd. at Ridgley Rd., (301) 561-2001

MASSACHUSETTS

BOSTON 730 Commonwealth Ave., (617) 739-1704, 111 Summer St., (617) 542-0361
BRAINTREE South Shore Plaza, 250 Granite St., (617) 848-9290
BROCKTON 675 Belmont, (617) 583-2270
BURLINGTON Crossroads Plaza, Rt. 3 S., (617) 229-2850
CAMBRIDGE Harvard Square, 28 Boylston St., (617) 354-7694
CHESTNUT HILL 200 Boylston St., (617) 969-2031
NATICK 1400 Worcester Rd., (617) 875-8721
SAUGUS 343 Broadway, (617) 233-4985
SPRINGFIELD 1985 Main St., Northgate Plz., (413) 732-4745
WORCESTER Lincoln Plaza, (617) 852-8844

MICHIGAN

ANN ARBOR 2515 Jackson Rd., (313) 761-6266
BIRMINGHAM 3620 W. Maple Rd., (313) 647-2151
DEARBORN Westborn Shop. Ctr., 23161 Michigan Ave (313) 274-6161
DETROIT DWNTN 1559 Woodward Ave., (313) 961-6855
FLINT G3298 Miller Rd., Yorkshire Plaza, (313) 732-2530
GRAND RAPIDS 3142 28th St. SE., (616) 957-2040
KALAMAZOO 25 Kalamazoo Center, (616) 343-0780
LANSING 2519 S. Cedar St., (517) 372-1120
PLAINFIELD North Kent Mall., (616) 364-7558
PONTIAC North Oak Plaza-Pontiac Mall, 2436 Elizabeth Lake Rd., (313) 682-6800
LIVONIA 33470 W. 7 Mile Rd., (313) 476-6800
ROSEVILLE 31873 Gratiot Ave., (313) 296-6210
SOUTHFIELD 17651 West 12 Mile Rd., (313) 569-1027
TROY Oakland Plaza, 322 John R. Rd., (313) 585-3900
WARREN 29038 Van Dyke Ave., (313) 573-3870

MINNESOTA

BLOOMINGTON 10566 France Ave. S., (612) 884-1641
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy., (612) 542-8471
ST. PAUL 6th & Wabasha, (612) 291-7230

MISSISSIPPI

JACKSON 979 Ellis Ave., (601) 352-5001

MISSOURI

DES-PERES 11960 Manchester Rd., (314) 965-5911
FLORISSANT 47 Florissant Oaks S/C, (314) 921-7722
INDEPENDENCE 1325 S. Noland Rd., (816) 254-3701
KANSAS CITY 4025 N. Oak Trafficway, (816) 455-3381
ST. ANN 10472 St. Charles Rock Rd., (314) 428-1400
SPRINGFIELD 2684 S. Glenstone, (417) 883-4320

NEBRASKA

LINCOLN 4601 "O" St., (402) 483-7841
OMAHA 3006 Dodge St., (402) 346-4003, 1318 72nd St. at Pacific, (402) 397-1086

NEVADA

LAS VEGAS Commercial Center, 953 E. Sahara #31-B, (702) 731-3956
RENO 3328 Kietzke Lane, (702) 826-6327

NEW HAMPSHIRE

MANCHESTER Hampshire Plaza, 1000 Elm St., (603) 625-4040
NASHUA 429 Amherst St., Rt. 101A, (603) 881-8588

NEW JERSEY

BRIDGEWATER 1472 U.S. Highway 22 East, (201) 469-3232
E. BRUNSWICK 595 A Rt. 18, (201) 238-7142
E. HANOVER Rt. 10, Hanover Plaza, (201) 884-1200
LAWRENCEVILLE Rt. 1 & Texas Ave., (609) 771-8113
NEWARK 595 Broad, (201) 622-1339
NORTHFIELD 322-24 Tilton Rd., (609) 645-7676
PARAMUS 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rt. #22 Center Isle, (201) 467-9827
VOORHEES 35 Eagle Plaza, (609) 346-0600

NEW MEXICO

ALBUQUERQUE 2108 San Mateo NE., (505) 265-9587

NEW YORK

ALBANY Shoppers Pk., Wolf Rd., (518) 459-5527
BAYSHORE 1751 Sunrise Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-6403
BROOKLYN 531 68th St., (212) 238-3576
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
FRESH MEADOWS 187-12 Horace Harding Exp., (212) 454-1075
GARDEN CITY 960 Franklin Ave., (516) 294-3350
JOHNSON CITY Giant Shopping Center, Harry L. Drive, (607) 729-6312
KINGSTON Kings Mall, Rt. 9W, (914) 336-6262
MELVILLE TSS Mall, Rt. 110, (516) 673-4646
NEWBURGH Zayre Plaza, Rt. #17K, (914) 561-2960
NEW ROCHELLE 242 North Ave., (914) 636-0700
NEW YORK 385 Fifth Ave., (212) 889-1345, 139 E. 42nd St., (212) 953-6053; 19 W. 23rd St., (212) 691-1861; 347 Madison Ave., (212) 867-8650; 270 Park Ave. South, (212) 533-0470; 1282 Broadway, (212) 279-9260; 9 Broadway, (212) 482-8200
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd., (716) 283-2041
REGO PARK 97-77 Queens Blvd., (212) 897-5200
ROCHESTER 3000 Winton Rd., (716) 244-6400
SCARSDALE 365 Central Park Ave., (914) 472-2520
SPRING VALLEY White House Center, 88 W. Rt. 59, (914) 425-2828
STATEN ISLAND 2409 Richmond Ave., (212) 698-3100
SYRACUSE 2544 Erie Blvd., (315) 446-3017; Hotel Syracuse, 510 S. Warren St., (315) 471-6663
UTICA Riverside Mall, (315) 735-1933
VALLEY STREAM Green Acres Shop. Ctr., (516) 872-6460

NORTH CAROLINA

CHARLOTTE 3732 Independence Blvd., (704) 535-6320; Tyvola Mall, 5401 South Blvd., (704) 527-3373
DURHAM South Square Mall, (919) 493-4025
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St., (919) 483-2418
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townridge Sq., Hwy. 70 W., (919) 781-9380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0030

OHIO

AKRON Fairlawn Plaza, 2727 W. Market St., (216) 836-9303
BEDFORD HEIGHTS 5217 Northfield Rd., (216) 662-2477
CANTON 5248 Dressler Rd. NW., (216) 494-7230; Mellet Plaza, 3826 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 793-8688; 16-18 Convention Way (on Skywalk), (513) 381-4664
CLEVELAND 419 Euclid (Dwntwn), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2805; The Patio Shop. Ctr., 4661 Karl Rd., (614) 436-4666; 400 N. High St., (614) 464-2781
DAYTON Northwest Plaza, 3279 West Siebenthaler, (513) 277-6500
ELYRIA 286 Midway Blvd., (216) 324-7741
FAIRFIELD 7255 Dixie Hwy. (1/4 Mi. North of I-275), (513) 874-5994
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 7551 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Central Ave., (419) 531-5797; Brownstone Plaza, 1724 S. Reynolds Rd., (419) 867-9153
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

OKLAHOMA

OKLAHOMA CITY 4732 SE 29th St., (405) 670-4561, Springdale S/C, 4469 NW 50th, (405) 943-8712, 1101 SW 59th St., (405) 634-2406
TULSA 7218 & 7220 E. 41st St., (918) 663-2190

OREGON

EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157, 9131 SE Powell, (503) 777-2223, 3rd and Washington Sts (Downtown), (503) 241-7024
SALEM Salem Plaza, 403 Center, (503) 588-7095

PENNSYLVANIA

ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 395-7155
BALA CYNWYD 67 E. City Line Ave., (215) 668-9950
EASTON 25th St. Shopping Center, (215) 250-0160
ELKINS PARK Elkins Park Square, 808 Old York Rd., (215) 645-5460
ERIE 5755 Peach St., (814) 868-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-5753
LANCASTER Park City Plaza, US 30, (717) 393-5817
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 362-1200
PHILADELPHIA 7542 Castor Ave., (215) 342-2217; 1002 Chestnut St., (215) 923-3080; 1801 Market St., 10 Penn Center, (215) 568-0901
YORK York County Shopping Center, (717) 757-2786
PITTSBURGH 5775 Baptist Rd., Hills Plaza, (412) 831-9694; 303 Smithfield St., (412) 391-3150; 4643 Baum Blvd., (412) 681-4000; 4768 McKnight Rd., (412) 364-4342
SCRANTON 206 Meadow Ave., (717) 348-1801
WYOMISSING Berkshire Mall West, 1101 Woodland Rd., (215) 372-8610

PUERTO RICO

HATO REY 243 Franklin D. Roosevelt Ave., (809) 759-8248

RHODE ISLAND

E. PROVIDENCE 850 Waterman Ave., (401) 438-2860
PROVIDENCE 177 Union St., (401) 831-0320

SOUTH CAROLINA

COLUMBIA Old Sears Bldg., 1001 Harden St., (803) 799-2065
GREENVILLE N. Hills S/C, (803) 292-1835
N. CHARLESTON 5900 Rivers Ave., (803) 747-5580

SOUTH DAKOTA

SIOUX FALLS 1700 S. Minnesota at 25th, (605) 331-3801

TENNESSEE

CHATTANOOGA 636 Northgate Mall, (615) 870-1366
JOHNSON CITY Peerless Center, (615) 282-6829
KNOXVILLE Cedar Bluff S/C, 9123 Executive Park Dr., (615) 690-0520
MEMPHIS 4665 American Way, (901) 795-4963; 1997 Union Ave., (901) 278-7935
NASHVILLE 2115 Franklin Pike, (615) 298-5484; Rivergate Plaza, (615) 859-3414

TEXAS

AMARILLO Wellington Sq. S/C, 1619 S. Kentucky, (806) 358-4567
ARLINGTON 2500 E. Randol Mill, Suite 113, (817) 274-3127
AUSTIN 8764 E. Research Blvd., (512) 459-4238; Southwood Mall, 1501 Ben White Blvd., (512) 447-0371
BROWNSVILLE 1639 Price Rd. (Hwy. 77), (512) 544-6800
BEAUMONT 5330 Eastex Frwy., (409) 898-7000
COLLEGE STATION 2414 Texas Ave., South, (409) 764-8171
CORPUS CHRISTI 1711 S. Staple St., (512) 887-8901
DALLAS 15340 Dallas Pkwy., Suite 1100, (214) 934-0275; 2930 W. Northwest Hwy., (214) 350-4144; 1517 Main St., (214) 760-8601, 2588 Royal Ln., (214) 484-9947
EL PASO 9515 Gateway West, (915) 594-8211, Kern Plaza Shopping Center, 3100 N. Mesa, (915) 544-8156
FT. WORTH 231 One Tandy Center, (817) 335-7198, 2801 Alta Mere, (817) 738-0251
HARLINGEN 1514 S. Hwy 77, Sunshine Strip, (512) 425-8890
HUMBLE 19300 "B" Hwy. 59 (at FM 1960), (713) 446-6314
HOUSTON 211C-FM 1960, (713) 444-7006; 10543 Gulf Fwy., (713) 943-9310; 5900 North Fwy., (713) 699-1932, 6813 SW Fwy., (713) 777-7907; 809 Dallas St., (713) 651-3002; Holland Square Center, 10920 East Freeway, (713) 453-0500; Champion Forest Plaza, Champion Forest Dr. and F.M. 1960 West, (713) 580-1404; 1018 Gessner, (713) 984-9600; 3278 South Loop West (So. Main at 610), (713) 661-5250
HURST Northeast Mall, (817) 284-1518
LAREDO 102 East Calton Rd., (512) 727-4768
LUBBOCK 3625 34th St., (806) 793-1467
ODESSA 1613 "A" East 8th Street, (915) 334-8355
RICHARDSON Fleetwood Sq. S/C, 202 W. Campbell Rd., (214) 669-1494
SAN ANTONIO 6018 West Ave., (512) 344-8792, 4249 Centergate, (512) 657-3958; Riverbend Parking Garage, 211 E. Market St. at Presa, (512) 225-0345

UTAH

OGDEN K-Mart Shopping Center, 3672 Wall Ave., (801) 394-1666
OREM Grand Central Plaza, 384 East & 1300 South, (801) 226-8651
MURRAY 6051 S. State Ave., (801) 268-6978
SALT LAKE CITY 301 South State St., (801) 322-5526

VIRGINIA

ALEXANDRIA 3425 King St. at Quaker Ln., (703) 820-2600
ARLINGTON Crystal City, 2301 So. Jefferson Davis Hwy., (703) 521-7479
FAIRFAX Westfair Center, 11027 Lee Hwy., (703) 273-6500
LYNCHBURG Hill's Plaza, Ward's Rd., (804) 237-1664
NEWPORT NEWS Newmarket South Shop. Ctr., (804) 245-8203
NORFOLK 5731 Poplar Hall Dr., (804) 461-0798; Wards Corner, 122 E. Little Creek Rd., (804) 588-7548
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453, 7728 Midlothian Turnpike, (804) 272-8803
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5955

WASHINGTON

BELLEVUE Crossroads Mall, North East 8th & 156 St., (206) 644-1804
BELLINGHAM 1111 Cornwall Ave., Suite B & C, (206) 676-0811
FEDERAL WAY 33505 Pacific Hwy. South, (206) 838-6830
OLYMPIA 106 N. Wilson, (206) 754-6111
SEATTLE 18405 Aurora Ave. N., (206) 542-6184, 1521 3rd Ave., (206) 447-1959; 5030 Roosevelt Way NE, (206) 527-1019
SPOKANE 7702 N. Division, (509) 484-7000; E. 12412 Sprague, (509) 922-2800
TACOMA 7030 S. Sprague, (206) 473-7333
TUKWILA 15425 53rd Ave. S., (206) 248-3710
YAKIMA 1111 N. First St., (509) 248-9667

WEST VIRGINIA

HURTINGTON 2701½ 5th Ave., (304) 523-3527

WISCONSIN

APPLETON 2310 West College Ave., (414) 731-6424
MADISON 57 West Towne Mall, (608) 833-6130
MILWAUKEE 6450 N. 76th St., (414) 353-6790
WEST ALLIS 2717 South 108th St., (414) 327-4240

Radio Shack®

TRS-80 Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910

FIRST CLASS
U.S. POSTAGE
PAID
Radio Shack
A Div. of Tandy Corp.

ADDRESS CHANGE
☐ **Remove from list** ☐ **Change as shown**
Please detach address label and mail to address shown above